

**SISTEM KENDALI PAGAR RUMAH BERBASIS *INTERNET OF THINGS*  
DENGAN PERINTAH SUARA MENGGUNAKAN ALGORITMA  
*BOYER-MOORE***

**SKRIPSI**

**Oleh :  
MUSTAFA KAMAL RABBANI  
NIM, 17650057**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**SISTEM KENDALI PAGAR RUMAH BERBASIS *INTERNET OF THINGS*  
DENGAN PERINTAH SUARA MENGGUNAKAN ALGORITMA  
*BOYER-MOORE***

**SKRIPSI**

**Diajukan Kepada :  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :  
MUSTAFA KAMAL RABBANI  
NIM. 17650057**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
202**

**LEMBAR PERSETUJUAN**  
**SISTEM KENDALI PAGAR RUMAH BERBASIS *INTERNET OF THINGS***  
**DENGAN PERINTAH SUARA MENGGUNAKAN ALGORITMA**  
***BOYER-MOORE***

**SKRIPSI**

**Oleh :**  
**MUSTAFA KAMAL RABBANI**  
**NIM, 17650057**

Telah diperiksa dan Disetujui untuk Diuji

Tanggal: 7 Juni 2021

Pembimbing I,




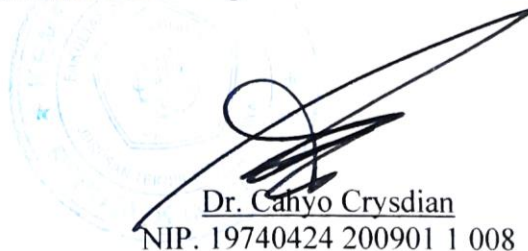
Ajib Hanani, M.T  
NIDT. 19840731 20160801 1 076

Pembimbing II,



Dr. Fachrul Kurniawan, M.MT  
NIP. 197710202009121001

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

**LEMBAR PENGESAHAN**  
**SISTEM KENDALI PAGAR RUMAH BERBASIS *INTERNET OF THINGS***  
**DENGAN PERINTAH SUARA MENGGUNAKAN ALGORITMA**  
***BOYER-MOORE***

**SKRIPSI**

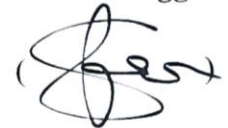
**Oleh :**  
**MUSTAFA KAMAL RABBANI**  
**NIM, 17650057**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Pada Tanggal : 7 Juni 2021


**Susunan Dewan Penguji**

- |                       |  |
|-----------------------|--|
| 1. Penguji Utama      | : <u>Dr. M. Amin Hariyadi, M.T</u><br>NIP. 19670018 200501 1 001 |
| 2. Ketua Penguji      | : <u>Juniardi Nur Fadila, MT</u><br>NIP. 19920605201903 1 015    |
| 3. Sekretaris Penguji | : <u>Ajib Hanani, M.T</u><br>NIDT. 19840731 20160801 1 076       |
| 4. Anggota Penguji    | : <u>Dr. Fachrul Kurniawan, M.MT</u><br>NIP. 197710202009121001  |

**Tanda Tangan**



Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

### PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Mustafa Kamal Rabbani

NIM : 17650057

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Sistem Kendali Pagar Rumah Berbasis *Internet of Things*  
Dengan Perintah Suara Menggunakan Algoritma  
*Boyer-Moore*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 7 Juni 2021  
Yang membuat pernyataan,



Mustafa Kamal Rabbani  
17650057

## HALAMAN MOTO



## **HAMALAN PERSEMBAHAN**

*Bismillahirrohmanirrohim*

*Alhamdulillah* puji syukur atas kehadiran Allah SWT, shalawat dan salam kepada Rasulullah SAW, penulis persembahkan sebuah karya ini kepada:

Kedua orang tua penulis yang sangat amat dicintai, Bapak Bambang Kuswanto dan Ibu Sumarlik yang selalu memberikan dukungan semangat dan doa.

Dosen pembimbing penulis Bapak Ajib Hanani, M.T dan Bapak Dr. Fachrul Kurniawan, M.MT yang telah memberikan bimbingan dan masukan dalam penelitian skripsi ini.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, serta seluruh guru – guru penulis yang telah membimbing dan memberikan ilmunya kepada saya.

Keluarga Teknik Informatika 2017 “Unocore” dan seluruh keluarga besar Teknik Informatika UIN Maulana Malik Ibrahim Malang.

Teman-teman komunitas “ONTAKI” yang telah memberikan kehangatan dan pengalaman serta ilmu yang sangat bermanfaat.

Orang-orang yang penulis sayangi, yang selalu memberikan semangat dan bantuan dalam menyelesaikan skripsi ini. Penulis ucapkan terimakasih yang sangat luar biasa.

## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

*Alhamdulillah* puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis mampu menyelesaikan skripsi dengan lancar dan tepat waktu. Sholawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW atas syafaatnya yang telah menuntun umat manusia menuju jalan yang baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan jazakumullah ahsanal jaza' kepada semua pihak yang telah membantu terselesaikannya Skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Orang tua dan keluarga penulis yang telah memberikan dukungan serta doa sehingga skripsi ini dapat terselesaikan.
2. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang dan jajarannya.
3. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang dan jajarannya.
4. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
5. Ajib Hanani, M.T, selaku Dosen Pembimbing I yang telah memberikan ilmu dan pengarahan dalam menyelesaikan skripsi ini.
6. Dr. Fachrul Kurniawan, M.MT, selaku Dosen Pembimbing II yang telah memberikan ilmu dan pengarahan dalam menyelesaikan skripsi ini.
7. Segenap sivitas akademi Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
8. Teman – teman Teknik Informatika 2017 “Unocore” yang telah memberikan semangat dan do'a kepada penulis.
9. Komunitas “ONTAKI” yang telah memberikan ilmu yang sangat bermanfaat.



10. Seluruh keluarga besar Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

11. Semua pihak yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Amin Ya Rabbal Alamin.*

*Wassalamu'alaikum Wr. Wb.*

Malang, 7 Juni 2021

Penulis

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
HALAMAN MOTO .....	v
HAMALAN PERSEMBAHAN .....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL .....	xiv
ABSTRAK .....	xv
ABSTRACT.....	xvi
الملخص.....	xvii
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang Penelitian .....	1
1.2. Pernyataan Masalah.....	4
1.3. Tujuan Penelitian.....	5
1.4. Batasan Penelitian .....	5
1.5. Manfaat Penelitian.....	5
BAB II .....	6
TINJAUAN PUSTAKA .....	6
2.1 Pencocokan Pattern ( <i>Pattern Matching</i> ) .....	6
2.1.1 <i>Boyer-Moore</i> .....	7
2.1.2 <i>Fase Preprocessing</i> .....	8

2.1.2.1 <i>Bad-character heuristic</i> .....	9
2.1.2.2 <i>Good-character heuristic</i> .....	11
2.1.3 Fase Pencarian .....	12
2.2 <i>Speech Recognition</i> .....	12
2.3 <i>Stemming</i> .....	13
<b>BAB III.....</b>	<b>14</b>
<b>METODOLOGI PENELITIAN .....</b>	<b>14</b>
3.1 Desain dan Implementasi .....	14
3.1.1 Akuisisi Data .....	15
3.1.2 Stemming .....	16
3.1.3 <i>Preprocessing Boyer-Moore</i> .....	16
3.1.4 Pencocokan <i>Boyer-Moore</i> .....	19
3.2 Perancangan Antarmuka ( <i>Interface</i> ) .....	20
3.3 Rangkaian Komponen .....	21
3.4 Uji Coba Sistem.....	22
<b>BAB IV .....</b>	<b>25</b>
<b>UJI COBA DAN PEMBAHASAN.....</b>	<b>25</b>
4.1. Implementasi Sistem .....	25
4.1.1 Spesifikasi <i>Alibaba Cloud ECS</i> .....	25
4.1.2 Halaman Aplikasi .....	25
4.1.3 Halaman <i>Speech Recognition</i> .....	26
4.2 Pengujian Sistem .....	27
4.3 Hasil Pengujian Sistem.....	27
4.3.1 Kendali dengan <i>button</i> .....	27
4.3.2 Kendali dengan <i>Speech Recognition</i> .....	29

4.3.3	Proses <i>PreBmBc</i> .....	30
4.3.4	Proses <i>PreBmGs</i> .....	31
4.3.5	Proses Pengujian <i>Boyer-Moore</i> .....	32
4.3.6	Pengujian Algoritma <i>Boyer-Moore</i> .....	32
4.4	Integrasi Islam .....	45
<b>BAB V.....</b>		<b>51</b>
<b>KESIMPULAN DAN SARAN .....</b>		<b>51</b>
5.1.	Kesimpulan .....	51
5.2.	Saran .....	51
<b>DAFTAR PUSTAKA .....</b>		<b>48</b>
<b>LAMPIRAN.....</b>		<b>50</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Mekanisme Sliding-Window .....	7
<b>Gambar 2.2</b> bad-character shift, x berada disebelah kiri P[j] .....	9
<b>Gambar 2.3</b> bad-character shift, x berada disebelah kanan P[j] .....	10
<b>Gambar 2.4</b> bad-character shift, x tidak terdapat pada P[j] .....	10
<b>Gambar 2.5</b> good-suffix shift, u terjadi lagi didahului karakter z berbeda dari y11	
<b>Gambar 2.6</b> good-suffix shift, hanya suffix dari u yang terjadi lagi di pattern P[j]	
.....	12
<b>Gambar 3.1</b> Desain Sistem .....	15
<b>Gambar 3.2</b> Flowchart Stemming.....	16
<b>Gambar 3.3</b> Penyelesaian Prosedur PreBmBc.....	17
<b>Gambar 3.4</b> Penyelesaian Prosedur PreBmGs.....	18
<b>Gambar 3.5</b> Penyelesaian Pengujian Boyer-Moore.....	19
<b>Gambar 3.6</b> Contoh Hasil Pengujian .....	20
<b>Gambar 3.7</b> Perancangan Antarmuka.....	21
<b>Gambar 3.8</b> Rangkaian Komponen .....	22
<b>Gambar 4.1</b> Halaman Utama .....	26
<b>Gambar 4.2</b> Fitur Speech Recognition .....	26
<b>Gambar 4.3</b> (a) Block code button off (b) Block code button on.....	28
<b>Gambar 4.4</b> Pesan yang diterima oleh subscribe .....	29
<b>Gambar 4.5</b> Block code Speech Recognition .....	29
<b>Gambar 4.6</b> Nilai OH pada masing-masing pattern .....	30
<b>Gambar 4.7</b> Nilai MH pada masing-masing pattern.....	31
<b>Gambar 4.8</b> Grafik waktu pengujian kalimat “Buka pagar rumah” .....	33
<b>Gambar 4.9</b> Grafik waktu pengujian kalimat “Tolong dibukakan pagar rumah”34	
<b>Gambar 4.10</b> Grafik waktu pengujian kalimat “Membuka pagar rumah” .....	35
<b>Gambar 4.11</b> Grafik waktu pengujian kalimat “Tutup pagar rumah” .....	36
<b>Gambar 4.12</b> Grafik waktu pengujian kalimat “Tolong ditutupkan pagar rumah”	
.....	38

<b>Gambar 4.13</b>	Grafik waktu pengujian kalimat “Menutup pagar rumah” .....	39
<b>Gambar 4.14</b>	Grafik waktu pengujian kalimat “Tolong pagar dibukakan” .....	40
<b>Gambar 4.15</b>	Grafik waktu pengujian kalimat “Pagar dibuka” .....	41
<b>Gambar 4.16</b>	Grafik waktu pengujian kalimat “Tolong pagar ditutupkan” .....	42
<b>Gambar 4.17</b>	Grafik waktu pengujian kalimat “Pagar ditutup” .....	43
<b>Gambar 4.18</b>	Grafik rata – rata waktu semua kalimat.....	44

## DAFTAR TABEL

<b>Tabel 3.1</b> Daftar pencacahan suffix dan prefix dari pattern. ....	18
<b>Tabel 4.1</b> Hasil pengujian kalimat “Buka pagar rumah” .....	33
<b>Tabel 4.2</b> Hasil pengujian kalimat “Tolong dibukakan pagar rumah” .....	34
<b>Tabel 4.3</b> Hasil pengujian kalimat “Membuka pagar rumah” .....	35
<b>Tabel 4.4</b> Hasil pengujian kalimat “Tutup pagar rumah” .....	36
<b>Tabel 4.5</b> Hasil pengujian kalimat “Tolong ditutupkan pagar rumah” .....	37
<b>Tabel 4.6</b> Hasil pengujian kalimat “Menutup pagar rumah” .....	38
<b>Tabel 4.7</b> Hasil pengujian kalimat “Tolong pagar dibukakan” .....	39
<b>Tabel 4.8</b> Hasil pengujian kalimat “Pagar dibuka” .....	40
<b>Tabel 4.9</b> Hasil pengujian kalimat “Tolong pagar ditutupkan” .....	41
<b>Tabel 4.10</b> Hasil pengujian kalimat “Pagar ditutup” .....	42

## ABSTRAK

Rabbani, Mustafa Kamal. 2021. **Sistem Kendali Pagar Rumah Berbasis *Internet of Things* Dengan Perintah Suara Menggunakan Algoritma *Boyer-Moore***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T., (II) Dr. Fachrul Kurniawan, M.MT.

Kata kunci: *Smart home, Boyer-Moore, Internet of Things, Pattern matching*

Perkembangan *Internet of Things* pada *smart home* saat ini tidak hanya dapat dikendalikan melalui tombol saja tetapi dapat dikendalikan juga melalui perintah suara dengan *Speech Recognition*. Penggunaan perintah dengan suara sendiri juga dapat mempermudah mobilitas pengguna dalam *smart home*. Pagar merupakan komponen keamanan yang sangat penting di rumah sebagai penjaga keamanan isi rumah dari bahaya kriminal dan pembatas jalan. Penelitian dilakukan menggunakan ESP8266 sebagai *microcontroller*, *Driver Stepper* dan *Motor Stepper* sebagai penggerak pagar rumah. Penelitian ini menggunakan algoritma *Boyer-Moore* sebagai *pattern matching*, penerapan algoritma *Boyer-Moore* dapat melakukan perintah tidak hanya dengan kalimat baku, tetapi juga dapat melakukan perintah dengan kalimat tidak baku dalam bahasa Indonesia yang sesuai dengan *pattern* yang sudah ditentukan. Berdasarkan penelitian yang sudah dilakukan, kalimat perintah yang diuji berjumlah 50 kalimat dan dilakukan pengujian masing – masing kalimat sebanyak 10 kali pengujian dengan 4 *pattern* (buka pagar, tutup pagar, pagar buka dan pagar tutup). Hasil pengujian mendapatkan akurasi sebesar 100 % dengan rata – rata waktu proses sekitar 0.00672 ms.



## ABSTRACT

Rabbani, Mustafa Kamal. 2021. **Fence Control System Based on Internet of Things using Voice Command of Boyer-Moore Algorithm.** Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisor: (I) Ajib Hanani, M.T., (II) Dr. Fachrul Kurniawan, M.MT.

Keywords: *Smart home, Boyer-Moore, Internet of Things, Pattern matching*

Currently, the development of the Internet of Things in smart homes are not merely controlled using a button but can be controlled through voice commands with speech recognition. The use of commands with a voice can also facilitate user mobility in the smart home. The fence is a very important security component in the house as a security guard for the contents of the house from criminal hazards and road dividers. This study was conducted using the ESP8266 as a microcontroller, driver stepper, and motor stepper as the driving force of the fence. This study uses the Boyer-Moore algorithm as pattern matching, the application using the Boyer-Moore algorithm can perform commands not only with the grammatical sentences but also can perform orders with a non-grammatical sentence in Indonesian that match a predetermined pattern. This study showed that the sentence commands tested are 50 sentences and each sentence is tested 10 times with 4 patterns (buka pagar, tutup pagar, pagar buka and pagar tutup). The test results get an accuracy of 100% with an average processing time of about 0.00672 ms.

## الملخص

رباني، مصطفى كمال. 2021. الإنترنت على أساس نظام مراقبة السياج المنزلي من الأشياء مع الأوامر الصوتية باستخدام خوارزمية *Boyer-Moore*. أطروحة. قسم المعلوماتية كلية الهندسة للعلوم والتكنولوجيا مولانا مالك إبراهيم الدولة الإسلامية جامعة مالانج. مستشار: (١) عجيب حناني، ماجستير.، (٢) فشرول كورنياوان، ماجستير.

الكلمات الرئيسية : المنزل الذكي، *Boyer-Moore*، إنترنت الأشياء، مطابقة النمط

تطوير إنترنت الأشياء على المنزل الذكي اليوم لا يمكن التحكم فيها فقط من خلال أزرار ولكن يمكن أيضا السيطرة عليها من خلال الأوامر الصوتية مع التعرف على الكلام. كما يمكن أن يؤدي استخدام الأوامر بصوتها الخاص إلى تسهيل تنقل المستخدمين في منزل ذكي. الأسوار هي عنصر أمني مهم جدا في المنزل كحارس أمن لمحتويات المنزل من المخاطر الجنائية وحواجز الطرق. تم إجراء البحث باستخدام ESP8266 كمتحكم دقيق وسائق السائر ومحرك السائر كمحرك سياج منزلي. تستخدم هذه الدراسة خوارزمية *Boyer-Moore* كنمط مطابق ، يمكن لتطبيق الخوارزمية *Boyer-Moore* تنفيذ الأوامر ليس فقط مع الجمل القياسية ، ولكن أيضا يمكن تنفيذ الأوامر مع الجمل غير القياسية باللغة الإندونيسية التي تتوافق مع النمط المحدد مسبقا. واستنادا إلى البحوث التي أجريت، بلغت جمل الأوامر التي تم اختبارها 50 جملة وأجرت اختبارات لكل منها - كل جملة تصل إلى 10 مرات اختبار مع 4 أنماط (سياج مفتوح، سياج وثيق، سياج مفتوح وغطاء سياج). حصلت نتائج الاختبار على دقة 100 ٪ مع متوسط وقت المعالجة حوالي 0.00672 ملي ثانية.

# BAB I

## PENDAHULUAN

### 1.1.Latar Belakang Penelitian

Pengaruh perkembangan teknologi sudah meliputi hampir keseluruhan bidang yang ada di bumi. Selalu terdapat teknologi yang baru, berkembang, berinovasi dan akan dibuat sangat canggih seiring berjalannya waktu. Hal tersebut dapat dibuktikan dengan maraknya inovasi baru yang dapat mempengaruhi gaya hidup dan pandangan manusia. Teknologi sendiri memiliki manfaat bagi manusia dengan memudahkan semua hal seperti perkembangan teknologi komunikasi dan informasi dapat mempermudah manusia dalam komunikasi dan memperoleh pengetahuan dengan mudah tanpa terkendala oleh jarak. *Smartphone* saat ini merupakan barang yang sudah menjadi kebutuhan bagi manusia. Hal tersebut dapat dibuktikan dengan perusahaan platform media sosial dari Kanada yaitu *Hootsuite* dan bekerjasama dengan *We are Social* dari Inggris yang merilis perkembangan pengguna internet di seluruh dunia, termasuk Indonesia dengan judul laporan "*Digital 2020: A comprehensive look at the state of the internet, mobile devices, social media, and e commerce*", pada tahun 2020 pengguna internet mencapai 175,4 juta jiwa dari total jumlah penduduk Indonesia yang mencapai 272,1 juta jiwa dan jumlah *smartphone* yang terkoneksi mencapai 338,2 juta unit, yang artinya hampir rata-rata orang Indonesia mempunyai lebih dari satu unit *smartphone*, sedangkan jumlah pengguna sosial media mencapai 160 juta jiwa (cyberthreat.id, 2020). Dengan jumlah sebesar itu, Indonesia dapat

dipastikan akan negara dengan pengguna *smartphone* terbesar keempat di dunia setelah Cina, India dan Amerika (kominfo.go.id, 2015).

Banyak hal yang dapat kita lakukan menggunakan *smartphone* dengan hanya menekan tombol. Selain *smartphone* digunakan sebagai alat komunikasi dan mencari informasi dengan mudah dan tanpa terkendala dengan jarak, salah satu contoh lainnya yaitu melakukan pengontrolan terhadap suatu perangkat atau peralatan. Contoh penerapan sistem tersebut sangat banyak digunakan sekarang yaitu *smart home* atau rumah pintar. Dimana dulu menghidupkan dan mematikan saklar masih menggunakan *remote*. Hal tersebut dirasa masih kuno dan belum mencerminkan perkembangan teknologi sekarang. Rumah pintar adalah salah satu topik menarik yang sedang trend saat ini. Pagar merupakan komponen keamanan yang sangat penting di rumah sebagai penjaga keamanan isi rumah dari bahaya kriminal dan pembatas jalan. Bahaya kriminal dapat berupa masuknya pencuri, percobaan sadap pesawat telepon, dan lain-lain (Utama, 2020), serta seringnya pemilik rumah yang lalai akan halnya dalam menutup dan mengunci pagar. Sebagaimana dijelaskan dalam Al Quran surah *Al-Baqarah* ayat 125:

وَإِذْ جَعَلْنَا الْبَيْتَ مَثَابَةً لِّلنَّاسِ وَأَمْنَا وَاتَّخِذُوا مِن مَّقَامِ  
إِبْرَاهِيمَ مُصَلًّى ۖ وَعَهِدْنَا إِلَىٰ إِبْرَاهِيمَ وَإِسْمَاعِيلَ أَنَّ طَهِّرَا بَيْتِيَ  
لِلطَّائِفِينَ وَالْعَاكِفِينَ وَالرُّكَّعِ السُّجُودِ

Artinya : Dan (ingatlah), ketika Kami menjadikan rumah itu (Baitullah) tempat berkumpul bagi manusia dan tempat yang aman. Dan jadikanlah

*sebahagian makam Ibrahim tempat shalat. Dan telah Kami perintahkan kepada Ibrahim dan Ismail: "Bersihkanlah rumah-Ku untuk orang-orang yang thawaf, yang i'tikaf, yang ruku' dan yang sujud".* Dalam tafsir Quraish Shibah (Surat Al-Baqarah Ayat 125, 2020) ayat ini dijelaskan sebagai berikut : Renungkanlah pula kisah pembangunan Rumah Allah (*Bayt Allah*) di Mekah oleh Ibrahim dan Ismail, anaknya. Di dalamnya terkandung pelajaran yang amat berharga bagi yang memiliki hati nurani bersih. Kami menjadikan rumah peribadatan itu sebagai rujukan bagi seluruh makhluk dan tempat perlindungan yang damai. Kami memerintahkan manusia untuk menjadikan lokasi berdirinya Ibrahim saat pembangunan rumah itu sebagai musala. Kami memerintahkan Ibrahim dan Ismâ'il untuk menjaga rumah itu dari segala unsur yang menodai kesuciannya, mempersiapkannya dengan baik untuk mereka yang tawaf, beriktikaf dan bersembahyang.

Pada *smarthome* saat ini sistem kendali tidak hanya berupa tombol saja sebagai inputan tetapi suara juga dapat digunakan sebagai inputan. Perintah suara yang diberikan oleh pengguna ditafsirkan oleh perangkat mobile menggunakan bahasa alami dan perangkat mobile bertindak sebagai konsol sentral (Rani, 2017). Salah satu faktor terpenting dalam *Speech Recognition* adalah pengujian dari suara terhadap pola kata. Pengujian pola banyak digunakan dalam berbagai program atau aplikasi yang digunakan sehari-hari, seperti fitur pencarian pada editor teks, *web search engine* seperti *Google*, analisis citra, bioinformatics seperti pengujian rantai asam amino rantai DNA (Nursyahrina, 2016), agar memperoleh hasil kecocokan yang benar maka perlu diselesaikan dengan

menggunakan algoritma *pattern matching*. Dalam pencarian *pattern* berupa *string*, maka diperlukan algoritma pengujian *string* yang dicari dengan semua *string* yang terdapat dalam dokumen teks tersebut. Pengujian *pattern* atau *pattern matching* adalah proses pencarian semua kemunculan *string* pendek  $P[0..n-1]$  yang disebut *pattern* di *string* yang lebih panjang  $T[0..m-1]$  yang disebut teks. (Kumara, 2008).

Dalam penelitian Junaidi, (2018) menjelaskan *pattern matching* dengan menggunakan algoritma paling sederhana adalah *bruteforce*, yaitu dengan pengujian dari kiri ke kanan atau *left-to-right*. Sebuah *pattern* akan dicocokkan karakter per karakter dalam sebuah *text*, dan ketika sebuah karakter dalam *pattern* yang ingin dicocokkan tidak sesuai dengan salah satu karakter dalam *text*, maka pencarian akan diulang dan dimulai pada huruf teks selanjutnya. Sedangkan algoritma *Boyer-Moore* adalah *pattern matching* dengan pencarian bersifat *right-to-left*, yakni pencarian dimulai dari huruf paling kanan pada *pattern*. Oleh karena itu penggunaan algoritma *Boyer-Moore* jauh lebih cepat daripada *bruteforce* karena pergeseran yang lebih efisien, dengan efisiensi *Boyer-Moore* lebih terlihat ketika menggunakan banyak variasi karakter dalam teks. Oleh karena itu penelitian yang diajukan berjudul “Sistem Kendali Pagar Rumah Berbasis *Internet of Things* dengan Perintah Suara Menggunakan Algoritma *Boyer-Moore*”.

## 1.2.Pernyataan Masalah

Berdasarkan latar belakang yang telah diuraikan pada sub bagian sebelumnya, maka dapat diambil pernyataan penelitian yaitu seberapa akurat algoritma

*Boyer-Moore* dalam pengenalan kalimat perintah terhadap kata kunci *pattern* pada pengendalian pagar menggunakan *Speech Recognition*.

### **1.3.Tujuan Penelitian**

Merujuk pada pernyataan masalah, maka tujuan yang diharapkan penelitian ini yaitu untuk mengetahui akurasi algoritma *Boyer-Moore* dalam pengenalan kalimat perintah terhadap kata kunci *pattern* pada pengendalian pagar menggunakan *Speech Recognition*.

### **1.4.Batasan Penelitian**

Batasan dalam penelitian adalah sebagai berikut :

1. Pagar yang digunakan bertipe *sliding*.
2. *Device* yang digunakan sebagai *Speech Recognition* yaitu *smartphone* berbasis Android.
3. Sistem hanya menerima masukan perintah suara dalam Bahasa Indonesia.
4. Kendali pagar hanya buka dan tutup

### **1.5.Manfaat Penelitian**

Adapun manfaat yang diperoleh dalam penelitian ini adalah memudahkan sistem dalam pengujian *text* terhadap *pattern* agar memperoleh *output* dari perintah yang diinginkan.

## BAB II

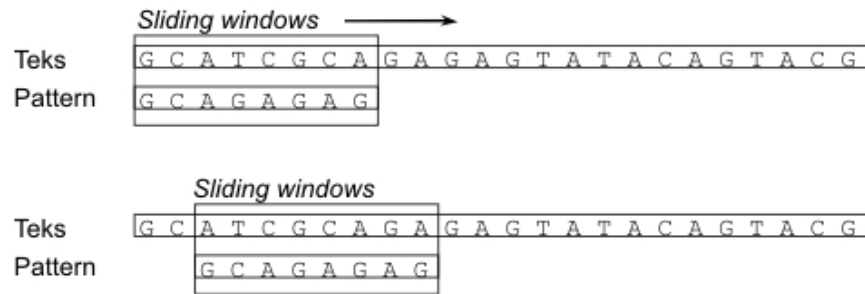
### TINJAUAN PUSTAKA

#### 2.1 Pencocokan Pattern (*Pattern Matching*)

Pengujian *pattern* atau *pattern matching* yang berupa *string* adalah proses pencarian jumlah kejadian kemunculan dari *pattern*  $P$  dengan panjang  $m$  terhadap *text*  $T$  dengan panjang  $n$ . Pengujian string atau string matching adalah proses pencarian semua kemunculan string pendek  $P[0..m-1]$  yang disebut pattern di string yang lebih panjang  $T[0..n-1]$  yang disebut teks. Pengujian string merupakan permasalahan paling sederhana dari semua permasalahan string lainnya, dan merupakan bagian dari pemrosesan data, pengkompresian data, lexical analysis, dan temu balik informasi. (Kumara, 2008).

Prinsip kerja dari algoritma *pattern matching* adalah, pertama memindai *text* dengan bantuan sebuah *window* yang ukuran panjangnya sama dengan Panjang *pattern*, lalu menempatkan *window* pada awal *text* agar sejajar, kemudian membandingkan karakter *text* pada *window* dengan karakter dari *pattern*, setelah pengujian dengan hasil cocok atau tidak cocok, maka dilakukan pergeseran *window* ke kanan. Proses ini dilakukan berulang-ulang sampai *window* berada di tepat akhir *text*. Mekanisme ini disebut *sliding-window* dapat dilihat pada Gambar 2.1 sebagai berikut.





**Gambar 2.1** Mekanisme Sliding-Window

Effendi (2013) menjelaskan algoritma *pattern matching* memiliki tiga komponen utama, pertama, *pattern* yaitu deretan karakter yang akan dicocokkan dengan *text*, yang dinyatakan dengan  $P[0..m-1]$  dengan panjang  $n$ , kedua, *text* yaitu tempat pengujian *pattern* dilakukan, yang dinyatakan dengan  $T[0..n-1]$  dengan panjang  $m$ , ketiga, alfabet yaitu karakter yang digunakan oleh bahasa dalam *pattern* dan *text*, dinyatakan dengan  $\Sigma$  dengan ukuran dinyatakan ASIZE.

### 2.1.1 Boyer-Moore

Algoritma *Boyer-Moore* adalah salah satu algoritma pencarian *string*, yang diterbitkan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1997. (Darmawan & Setianingrum, 2018). Algoritma *Boyer-Moore* telah dibuktikan sebagai salah satu algoritma dalam *pattern matching* paling efisien dalam aplikasi pencarian string dengan menggunakan *natural language*. Pada dasarnya algoritma ini memiliki kesamaan dengan algoritma *Knuth-Morris-Pratt* (KMP) dalam segi cara kerjanya dimana kedua algoritma ini melakukan lompatan pengecekan dalam proses pencarian *string*. Namun berbeda dengan algoritma

*Knuth-Morris-Pratt* (KMP), algoritma *Boyer-Moore* melakukan perbandingan *pattern* dimulai dari kanan ke kiri atau *right-to-left*, sedangkan algoritma *Knuth-Morris-Pratt* (KMP) melakukan perbandingan *pattern* dimulai dari kiri ke kanan atau *left-to-right*. Dalam algoritma *Boyer-Moore* menggunakan dua *heuristics* untuk memutuskan seberapa jauh lompatan agar mendapat hasil pengujian yang pas yaitu : *bad-character heuristic* atau juga bisa disebut *occurrence heuristic (OH)*, dan *good-character heuristic* atau juga disebut *match heuristic (MH)*. Informasi untuk heuristik setiap dipertahankan dalam sebuah array yang dibangun pada awal operasi yang cocok. (Orwant, 1999). Nursyahrina (2016) melakukan penelitian perbandingan perhitungan pengujian pola atau *pattern matching* algoritma *Boyer-Moore* dengan algoritma *Knuth-Morris-Pratt* (KMP) menggunakan aplikasi android *Google Now* memperoleh hasil dimana algoritma *Boyer-Moore* membutuhkan 11 kali pengujian sampai ditemukannya *pattern* dalam *text* lebih cepat dibanding algoritma *Knuth-Morris-Pratt* (KMP) dengan 21 kali pengujian. Pada algoritma *Boyer-Moore* terbagi atas 2 fase, yaitu Fase *Preprocessing* dan Fase Pencarian.

### **2.1.2 Fase *Preprocessing***

Fase *Preprocessing* adalah perpaduan dari perhitungan fungsi pergeseran dari *bad-character heuristic* dan *good-character heuristic* dimana nilai pergeseran disimpan pada tabel *BmBc* (Boyer Moore *bad-character*) dari pergeseran *bad-character heuristic* dan tabel *BmGs* (Boyer Moore *good-suff*) dari pergeseran *good-character heuristic*. Kedua fungsi tersebut digunakan dalam fase *Preprocessing* algoritma *Boyer-Moore*.

### 2.1.2.1 Bad-character heuristic

*Bad-character heuristic* adalah konsep pergeseran karakter pada *text* yaitu  $T[i]$  yang tidak cocok dengan *pattern*  $P[j]$  atau disebut **preBmBc**. Fungsi dari prosedur ini adalah untuk menentukan berapa besar pergeseran yang dibutuhkan untuk mencapai karakter tertentu yang berada pada *pattern* dari karakter *pattern* terkanan. Hasil dari pergeseran akan disimpan pada tabel *BmBc* yang berisi nilai *occurrence heuristic* (*OH*).

Konsep dari *bad-character heuristic* sebagai berikut:

1. Jika *mismatch* dari karakter  $T[i]$  terhadap *pattern*  $P[j]$  dan karakter  $T[i]$  adalah  $x$  yang dimana terletak lebih kiri dari  $P[j]$ , maka *pattern* digeser ke kanan agar posisi  $x$  di  $T[i]$  sejajar dengan posisi kemunculan terakhir atau *last occurrence* ( $lo$ ) yaitu  $x$  di *pattern*, pemeriksaan berikutnya tetap dimulai dari *index* terakhir  $P[j]$ , maka geser posisi *index*  $T[i]$  sejauh:

$$i_{new} = i + (m - 1) - lo = i + m - (lo + 1) \quad (2.1)$$

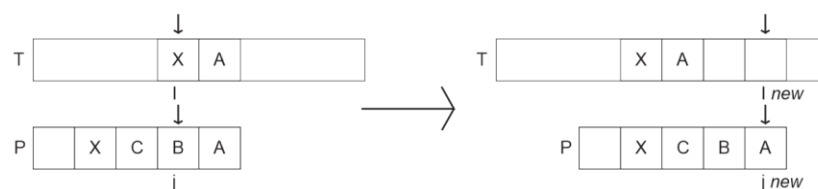
Keterangan :

$i_{new}$  = *index* baru

$m$  = panjang *pattern*

$i$  = *index* lama

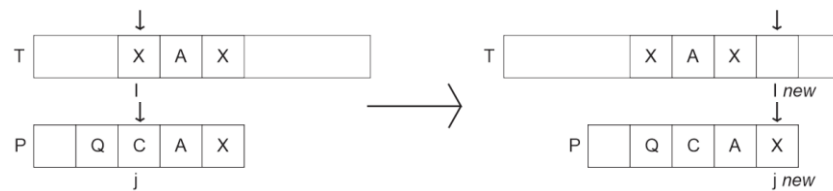
$lo$  = *last occurant* karakter



**Gambar 2.2** *bad-character shift*,  $x$  berada disebelah kiri  $P[j]$

2. Jika *mismatch* dari karakter  $T[i]$  terhadap *pattern*  $P[j]$  dan karakter  $T[i]$  adalah  $x$  yang dimana terletak lebih kanan dari  $P[j]$ , maka geser *pattern* satu karakter ke kanan agar posisi *index* terakhir *pattern* sejajar dengan posisi akhir *text* sebelumnya ditambah 1, pemeriksaan berikutnya tetap dimulai dari *index* terakhir  $P[j]$ , maka geser posisi *index*  $T[i]$  sejauh:

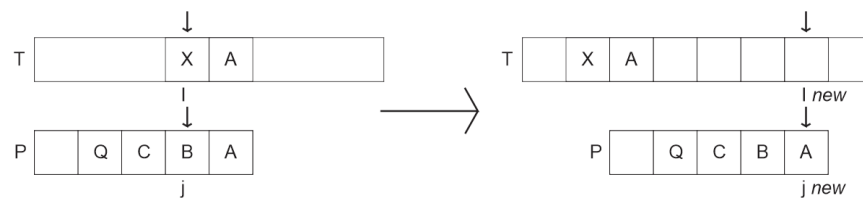
$$i_{new} = i + m - j \quad (2.2)$$



**Gambar 2.3** *bad-character shift*,  $x$  berada disebelah kanan  $P[j]$

3. Jika *mismatch* dari karakter  $T[i]$  terhadap *pattern*  $P[j]$  dan karakter  $T[i]$  adalah  $x$  yang tidak muncul pada *pattern*, maka geser  $P[j]$  agar posisi *index* pertama *pattern* atau  $P[0]$  sejajar dengan *index*  $i+1$ , pemeriksaan berikutnya tetap dimulai dari *index* terakhir  $P[j]$ , maka geser posisi *index*  $T[i]$  sejauh:

$$i_{new} = i + m \quad (2.3)$$



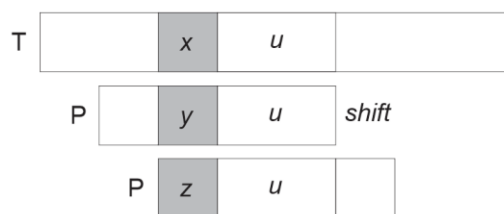
**Gambar 2.4** *bad-character shift*,  $x$  tidak terdapat pada  $P[j]$

### 2.1.2.2 Good-character heuristic

*Good-character heuristic* adalah konsep pergeseran karakter pada *text* yaitu  $T[i]$  yang tidak cocok dengan *pattern*  $P[j]$  atau disebut **preBmGs**. Prosedur *suffix* dijalankan terlebih dahulu terhadap *pattern*. Prosedur *suffix* adalah pemeriksaan kecocokan sejumlah karakter dimulai dari kanan *pattern* dengan jumlah karakter setiap karakter yang dimulai dari kiri *pattern*. Hasil dari prosedur ini akan disimpan pada tabel *suff*. Prosedur **preBmGs** menggunakan tabel *suff* untuk mengetahui semu pasangan segmen yang sama terhadap *pattern*. Hasil dari prosedur **preBmGs** disimpan pada tabel *BmGs* yang berisi nilai dari *match heuristic* (MH).

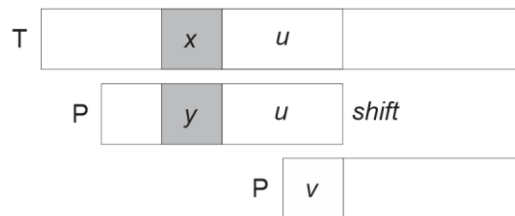
Konsep dari *bad-character heuristic* sebagai berikut:

1. Jika *mismatch* dari karakter  $T[i]$  terhadap *pattern*  $P[j]$ , maka geser  $P[j]=x$  ke karakter lain yang letaknya lebih kiri dari  $P[j]$  dan terletak di sebelah kiri segmen  $u$ .



**Gambar 2.5** *good-suffix shift*,  $u$  terjadi lagi didahului karakter  $z$  berbeda dari  $y$

2. Jika *mismatch* dari karakter  $T[i]$  terhadap *pattern*  $P[j]$  dan tidak ada segmen yang sama dengan  $u$ , maka cari  $u$  yang merupakan *suffix* terpanjang  $u$ .



**Gambar 2.6** *good-suffix shift*, hanya *suffix* dari *u* yang terjadi lagi di pattern  $P[j]$

### 2.1.3 Fase Pencarian

Fase pencarian adalah pergeseran berdasarkan perbandingan terbesar nilai  $BmBc$  dengan  $BmGs$ . Perbandingan dimulai dari karakter paling kanan dari *pattern*. Jika nilai kecocokan sama dengan Panjang *pattern* maka *pattern* telah ditemukan pada *text*, lalu geser *pattern* sejumlah panjang *pattern* ke arah kiri sampai awal *text* untuk melanjutkan pengujian selanjutnya.

## 2.2 *Speech Recognition*

*Speech Recognition* adalah proses identifikasi suara berdasarkan kata yang diucapkan dengan cara melakukan konversi sinyal akustik yang ditangkap oleh *audio device* atau proses untuk identifikasi suara seseorang berdasarkan dari kalimat atau kata yang diucapkan (Dirmansyah & Hermanto, 2014). *Speech Recognition* yang digunakan oleh penulis adalah *Google Speech Recognition* dimana digunakan untuk mengubah suara menjadi teks yang dimana aplikasi akan menggunakan hasil tersebut sebagai *text* yang akan dibandingkan dengan *pattern* yang tersedia. Dengan menggunakan *Google Speech Recognition* akan mempermudah pengguna untuk mengontrol pagar dengan cara pengguna cukup mengucapkan perintah.

### 2.3 Stemming

*Steeming* merupakan suatu proses menemukan kata dasar atau *root word* dari suatu kata yang berimbuhan atau *affixed word* dengan cara menghilangkan semua imbuhan atau *affix* yang terdiri dari awalan (*prefix*), akhiran (*suffix*), sisipan (*infix*) dan kombinasi kombinasi dari awalan dan akhiran atau *confix*. Dalam penelitian Wibowo (2016) menjelaskan bahwa proses *steaming* ini merupakan cara untuk meningkatkan performa *Information Retrivel (IR)*.

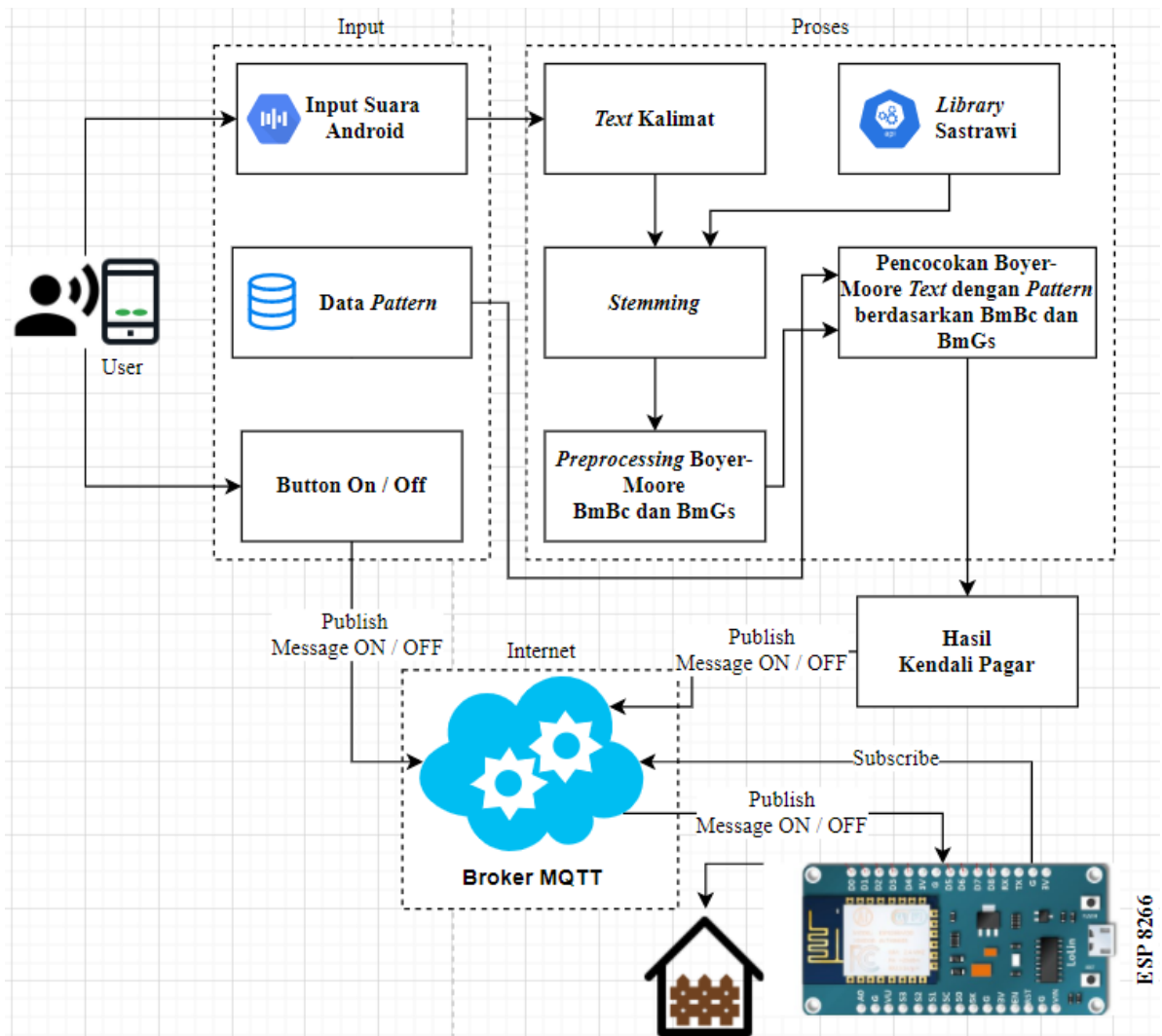
## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Desain dan Implementasi

Desain dan implementasi sistem ini terdiri dari *input*, proses, *output*, pengujian sistem dan rancangan *user interface*. Tahap *input* merupakan tahap akuisisi data yang kemudian menjadikan data yang siap diproses pada tahap proses, *user* dapat melakukan *input* dengan suara maupun *button*. Jika *user* menggunakan *input* suara yang artinya menggunakan fitur *Google Speech Recognition* dalam bentuk *text* lalu akan diproses oleh sistem, jika *user* menggunakan *input button* maka *output* akan langsung ke hasil kendali pagar dengan sesuai *button* yang di tekan yaitu buka pagar atau tutup pagar. Tahap proses adalah tahap hasil dari proses *Google Speech Recognition* dalam bentuk *text* kemudian di proses *Stemming* menggunakan *Library Sastrawi* lalu dilakukan proses *pattern matching* dengan menggunakan algoritma *Boyer-Moore* untuk memperoleh hasil pengujian yang diinginkan. Pada sistem pengujian akurasi metode *Boyer-Moore* dilakukan menggunakan *Confussion Matrix*. Desain sistem sistem ditunjukkan pada Gambar 3.1.





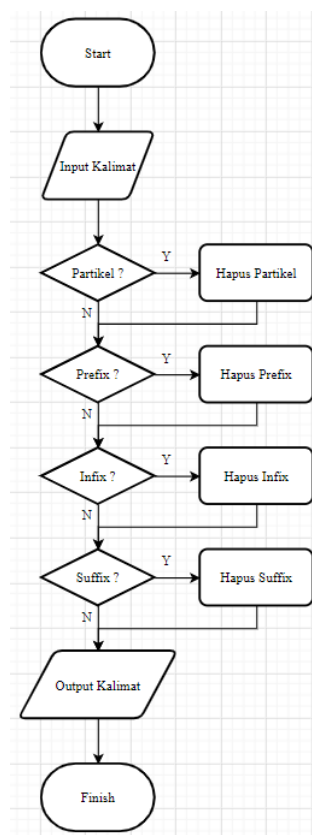
**Gambar 3.1** Desain Sistem

### 3.1.1 Akuisisi Data

Data yang digunakan dalam penelitian ini terdapat dua macam, yaitu data primer dan data sekunder. Dimana data primer didapatkan melalui inputan suara dari *Speech Recognition* dengan menggunakan *Google Speech* dalam Bahasa Indonesia dengan hasil format *text*. Data sekunder dibuat oleh peneliti sebagai data *pattern* yang digunakan sebagai kunci dari pengujian terhadap *text* yaitu buka pagar, tutup pagar, pagar buka dan pagar tutup.

### 3.1.2 Stemming

*Stemming* merupakan proses mengubah kalimat atau kata yang memiliki imbuhan menjadikan kata atau kalimat yang baku jadi kata atau kalimat tersebut yang akan diproses pada sistem adalah kata-kata dasar saja dalam Bahasa Indonesia. Proses *steeming* ini peneliti menggunakan *library* Sastrawi dengan bahasa pemrograman php. Alur *Stemming* ditunjukkan pada Gambar 3.2.



**Gambar 3.2** Flowchart Stemming

### 3.1.3 Preprocessing Boyer-Moore

*Boyer-Moore* memiliki tahap *Preprocessing* terdiri dari ***preBmBc*** dan ***preBmGs***. Prosedur ***preBmBc*** memiliki nilai *Occurrence Heuristic (OH)*. Langkah-langkah pada proses nya, pertama membuat *stack* dari *BmBc* dan diberi

nilai kosong, lalu melakukan perhitungan terhadap panjang *pattern* yang akan dimasukan nilai OH kedalam *stack BmBc* yang merujuk rumus 2.1.

Contoh kasus *Pattern* : buka

$$i_{new K} = i + (m - 1) - lo = i + (4 - 1) - 2 = i + 1$$

Pattern	B	U	K	A
Pergeseran			1	

Stack BmBc	Karakter	K
	Nilai OH	1

$$i_{new U} = i + (m - 1) - lo = i + (4 - 1) - 1 = i + 2$$

Pattern	B	U	K	A
Pergeseran		2		

Stack BmBc	Karakter	K	U
	Nilai OH	1	2

$$i_{new B} = i + (m - 1) - lo = i + (4 - 1) - 0 = i + 3$$

Pattern	B	U	K	A
Pergeseran	3			

Stack BmBc	Karakter	K	U	B
	Nilai OH	1	2	3

$$i_{new A} = i + (m - 1) - lo = i + (4 - 1) - 3 = i + 0$$

Pattern	B	U	K	A
Pergeseran				0

Stack BmBc	Karakter	K	U	B	A
	Nilai OH	1	2	3	0

Pattern	B	U	K	A
Pergeseran	3	2	1	0

Stack BmBc	Karakter	K	U	B	A
	Nilai OH	1	2	3	0

**Gambar 3.3** Penyelesaian Prosedur *PreBmBc*

Pada Gambar 3.3 dijelaskan bahwa karakter yang di pilih akan ditambahkan ke dalam *stack BmBc* dengan nilai OH sesuai pergeseran. Prosedur **preBmGs** memiliki nilai *Match Heuristic (MH)*. Langkah-langkah pada prosesnya yaitu, pertama membuat *stack* dari *BmGs* dengan nilai kosong dan tabel *Compare* berisi sejumlah karakter sebelah kanan *patter* dari pergeseran kanan ke kiri, lalu cacah *pattern* kemudian simpan pada tabel *Compare*. Kemudian cacah *prefix* dan *suffix* dari *pattern* kemudian simpan hasil *prefix* pada tabel *stack BmGs* dan bandingkan hasil *suffix* terhadap *Compare*.

Contoh kasus *Pattern* : buka

**Tabel 3.1** Daftar pencacahan *suffix* dan *prefix* dari *pattern*.

<i>Suffix</i> (Kanan ke Kiri)	<i>Prefix</i>	<i>Suffix</i> (Kiri ke kanan)
Null	B	UKA
B	U	KA
BU	K	A
BUK	A	Null

Dari Tabel 3.1 dijelaskan bahwa telah terjadi pemotongan dari kedua sisi dari *pattern* sehingga terbentuknya *suffix* dan *prefix*, nilai *null* diartikan kosong.

<i>Stack Compare</i>				
<i>Compare</i>	BUK	BU	B	Null
Pergeseran	1	2	3	4

Pattern	B	U	K	A
<i>Stack BmGs</i>				
Karakter	B	U	K	A
Nilai MH	-	-	-	1

MH awal selalui bernilai 1

Pattern	B	U	K	A
<i>Stack BmGs</i>				
Karakter	B	U	K	A
Nilai MH	-	-	4	1

<i>Prefix</i>	K
<i>Suffix</i>	A

*Suffix Compare* : Null, Pergeseran *Suffix Compare* ke 4 berdasarkan *Stack Compare*

Pattern	B	U	K	A
<i>Stack BmGs</i>				
Karakter	B	U	K	A
Nilai MH	-	4	4	1

<i>Prefix</i>	U
<i>Suffix</i>	KA

*Suffix Compare* : Null, Pergeseran *Suffix Compare* ke 4 berdasarkan *Stack Compare*

Pattern	B	U	K	A
<i>Stack BmGs</i>				
Karakter	B	U	K	A
Nilai MH	4	4	4	1

<i>Prefix</i>	B
<i>Suffix</i>	UKA

*Suffix Compare* : Null, Pergeseran *Suffix Compare* ke 4 berdasarkan *Stack Compare*

**Gambar 3.4** Penyelesaian Prosedur *PreBmGs*

Pada Gambar 3.4 dijelaskan bahwa penentuan nilai dari MH ditentukan dari pergeseran pada *Compare*, nilai MH selalu diberikan kepada semua karakter yang ada pada *pattern* dan penyusunan *stack BmGs* berdasarkan urutan *index* pada *pattern*.

### 3.1.4 Pencocokan *Boyer-Moore*

Pada tahap ini pengujian dimulai dari *index* paling kiri pada *text* dan pengujian per karakter dimulai paling kanan pada *pattern*, lalu setiap kali ditemukan ketidakcocokan maka bandingkan nilai OH dan MH pada *stack BmBc* dan *BmGs* yang sudah dibuat pada tahap *Preprocessing* sebelumnya.

Contoh kasus *Pattern* : buka, *Text* : membuka

<i>Stack BmBc</i>					<i>Stack BmGs</i>				
Karakter	B	U	K	A	Karakter	B	U	K	A
Nilai OH	3	2	1	0	Nilai MH	4	4	4	1

<i>Index</i>	0	1	2	3	4	5	6
<i>Text</i>	M	E	M	B	U	K	A
<i>Pattern</i>	B	U	K	A			

Max  $BmBc[B] = 3$  banding  $BmGs[A] = 1$   
Shift = 3 Pergeseran

<i>Index</i>	0	1	2	3	4	5	6
<i>Text</i>	M	E	M	B	U	K	A
<i>Pattern</i>				B	U	K	A

**Gambar 3.5** Penyelesaian Pengujian *Boyer-Moore*

Pada proses pencarian diatas memberikan hasil bahwa kata buka ditemukan pada *text* sehingga *Motor Stepper* akan bergerak untuk membuka pagar, begitu juga ketika kata tutup ditemukan pada *text* maka *Motor Stepper* akan bergerak untuk membuka pagar.

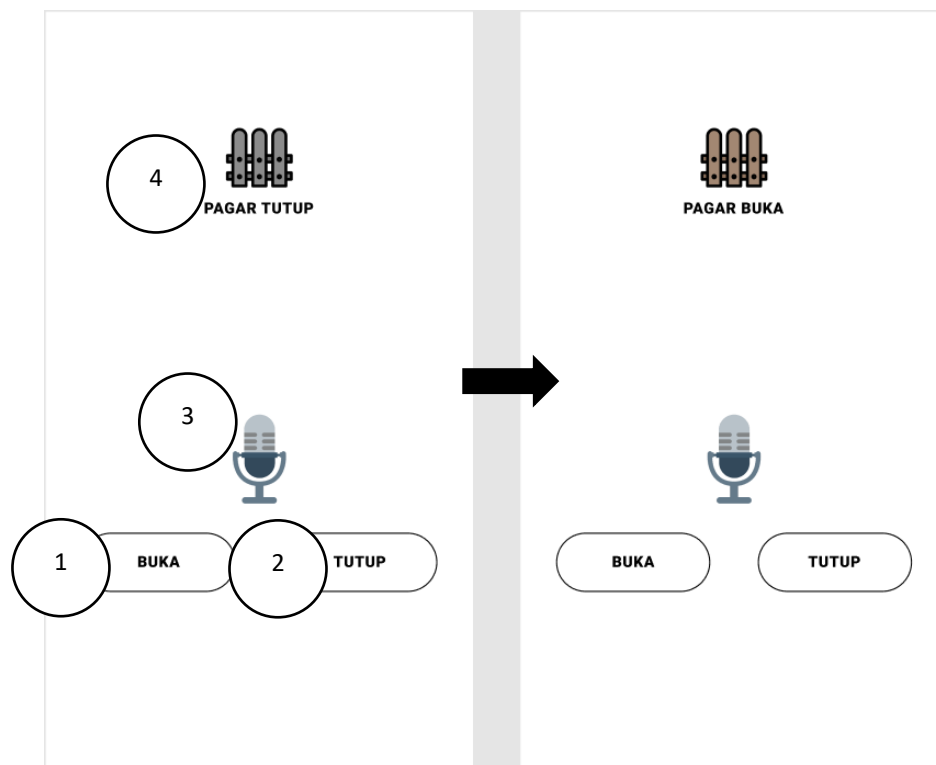
Text Asli: tolong bukakan pagar  
Text Stemming: tolong buka pagar  
Pattern : buka pagar  
Ditemukan pada posisi pergeseran ke : 7  
Status : buka  
motor maju

**Gambar 3.6** Contoh Hasil Pengujian

Pada Gambar 3.6 merupakan hasil dari pengujian dan terlihat bahwa pergeseran pada *pattern* tergantung dari hasil keputusan perbandingan nilai OH dan MH.

### **3.2 Perancangan Antarmuka (*Interface*)**

Proses perancangan antarmuka (*Interface*) pada penelitian ini dirancang agar dapat memberikan kemudahan dalam membangun sistem sehingga dapat dipahami tujuan dan fungsi dari setiap tampilan pada sistem tersebut. Pada Gambar 3.7 adalah tampilan awal *user*.



**Gambar 3.7** Perancangan Antarmuka

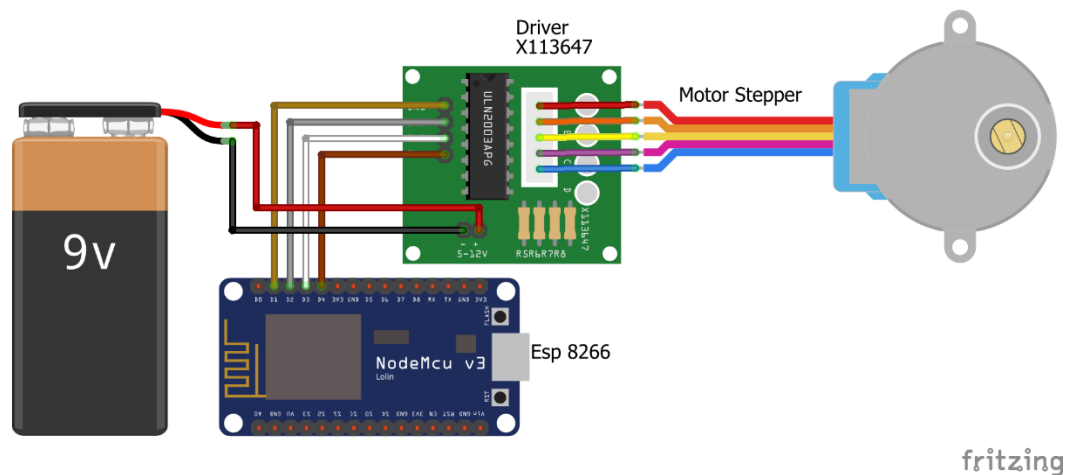
Keterangan Gambar :

1. *Button* Buka digunakan untuk membuka pagar secara manual,
2. *Button* Tutup digunakan untuk menutup pagar secara manual,
3. *Button Speech Recognition* digunakan untuk kendali pagar dengan menggunakan suara.
4. *ImageView* untuk menampilkan status pada pagar.

### 3.3 Rangkaian Komponen

Rangkaian komponen elektronik ini berperan sebagai *blueprint* dalam membangun sistem. Dalam penelitian ini menggunakan ESP8266 sebagai *microcontroller*, *Driver Stepper*, *Motor Stepper* sebagai penggerak dan Baterai

9V sebagai *power*, gambar rangkaian komponen dapat dilihat pada Gambar 3.8 sebagai berikut.



**Gambar 3.8** Rangkaian Komponen

### 3.4 Uji Coba Sistem

Pengujian sistem dilakukan untuk memperoleh hasil apakah sistem yang dibangun sudah akurat atau tidak, untuk mengukur algoritma *Boyer-Moore* perlu digunakan *Confussion Matrix*. *Confussion Matrix* terdiri dari *True positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*. Dalam penelitian Prakasa & Lhaksamana, (2018) menjelaskan bahwa *True positive* merepresentasikan data yang berada pada kelas positif yang diprediksi secara benar oleh algoritma. *False Positive* merepresentasikan data yang seharusnya berada pada kelas positif diprediksi menjadi kelas negatif oleh algoritma. *False Negative* merupakan data yang seharusnya berada di kelas *negative*. diprediksi menjadi kelas positif oleh algoritma. *True Negative* merupakan data yang berada pada kelas *negative* dan diprediksi secara benar oleh algoritma.



$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

Akurasi adalah proses matematika yang sering digunakan untuk mengevaluasi kinerja suatu algoritma. Akurasi dihitung berdasarkan rasio jumlah data yang diprediksi oleh algoritma dengan benar dan jumlah semua data dalam kumpulan data (Prakasa & Lhaksamana, 2018). Adapun daftar kalimat dan *pattern* yang akan diuji yaitu 50 kalimat perintah dengan 4 *pattern* (buka pagar, tutup pagar, pagar buka, pagar tutup) sebagai berikut:

Kalimat yang cocok dengan *pattern* buka pagar

- |                                |                                 |
|--------------------------------|---------------------------------|
| - Buka pagar                   | - Tolong bukakan pagar rumah    |
| - Tolong buka pagar            | - Tolong membuka pagar          |
| - Buka pagar rumah             | - Membuka pagar rumah           |
| - Tolong buka pagar rumah      | - Tolong membuka pagar rumah    |
| - Tolong dibuka pagar          | - Tolong membukakan pagar       |
| - Dibuka pagar rumah           | - Membukakan pagar rumah        |
| - Tolong dibuka pagar rumah    | - Tolong membukakan pagar rumah |
| - Tolong dibukakan pagar       | - Tolong bukakan pagar          |
| - Dibukakan pagar rumah        | - Bukakan pagar rumah           |
| - Tolong dibukakan pagar rumah |                                 |

Kalimat yang cocok dengan *pattern* tutup pagar

- |                      |                               |
|----------------------|-------------------------------|
| - Tutup pagar        | - Tolong tutupkan pagar       |
| - Tolong tutup pagar | - Tutupkan pagar rumah        |
| - Tutup pagar rumah  | - Tolong tutupkan pagar rumah |

- Tolong tutup pagar rumah
- Tolong ditutup pagar
- Ditutup pagar rumah
- Tolong ditutup pagar rumah
- Tolong ditutupkan pagar
- Ditutupkan pagar rumah
- Tolong ditutupkan pagar rumah
- Tolong menutup pagar
- Menutup pagar rumah
- Tolong menutup pagar rumah
- Tolong menutupkan pagar
- Menutupkan pagar rumah
- Tolong menutupkan pagar rumah

Kalimat yang cocok dengan *pattern* pagar buka

- Pagar buka
- Tolong pagar dibuka
- Tolong pagar buka
- Pagar dibukakan
- Pagar dibuka
- Tolong pagar dibukakan

Kalimat yang cocok dengan *pattern* pagar tutup

- Pagar buka
- Tolong pagar dibuka
- Tolong pagar buka
- Pagar dibukakan
- Pagar dibuka
- Tolong pagar dibukakan

## **BAB IV**

### **UJI COBA DAN PEMBAHASAN**

#### **4.1. Implementasi Sistem**

Implementasi sistem merupakan tahap untuk mempresentasikan hasil dari pengembangan perangkat lunak dari sistem perancangan yang telah dibangun dan nantinya untuk menjadikan sistem diuji dan siap digunakan. Peneliti menggunakan *Alibaba Cloud ECS* sebagai *Cloud Computing* digunakan untuk *Broker MQTT Mosquitto*, pembangunan sistem *Application* menggunakan *App Inventor* dan ESP8266 sebagai *microcontroller*.

##### **4.1.1 Spesifikasi *Alibaba Cloud ECS***

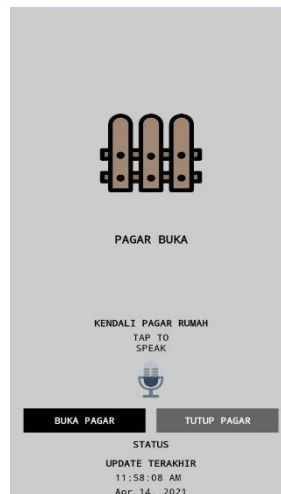
Spesifikasi yang digunakan pada *Alibaba Cloud ECS* yang digunakan dalam pembangunan sistem sebagai berikut:

1. CPU and Memory : 1Cores 1 GiB
2. Operating System : Ubuntu 20.04 64-bit
3. Disk Size : 40 GB
4. Region Zone : Malaysia (Kuala Lumpur Zone A)

##### **4.1.2 Halaman Aplikasi**

Pada halaman aplikasi merupakan tampilan utama pada sistem yang telah dibangun, pada halaman ini terdapat beberapa *button* pilihan, diantaranya *button* buka pagar, *button* tutup pagar dan *button Speech Recognition*. Pada halaman utama ini juga terdapat informasi terkait kondisi pagar saat ini sedang terbuka

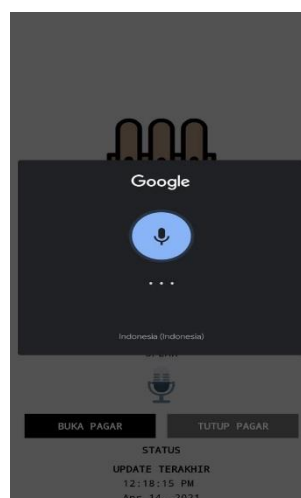
atau sedang tertutup, serta terdapat informasi terkait *update* terakhir terkait kondisi pagar. Halaman aplikasi dapat dilihat pada Gambar 4.1.



**Gambar 4.1** Halaman Utama

#### 4.1.3 Halaman *Speech Recognition*

Pada halaman ini terdapat fitur *speech recognition* yang merupakan proses *input* kata dalam Bahasa Indonesia menggunakan suara, pada fitur ini peneliti menggunakan *Google API Speech Recognition*. Halaman ini dapat dilihat pada Gambar 4.2.



**Gambar 4.2** Fitur *Speech Recognition*

## 4.2 Pengujian Sistem

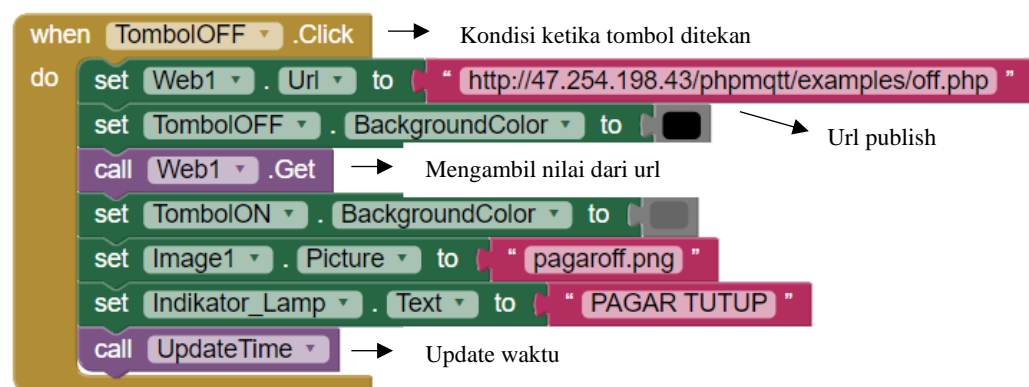
Pada tahap pengujian sistem ini dilakukan untuk mengetahui bagaimana sistem ini dibuat agar berjalan dengan lancar dan bagaimana proses pengujian kalimat perintah terhadap *pattern* menggunakan algoritma *Boyer-Moore* sudah berjalan sesuai dengan perancangan.

## 4.3 Hasil Pengujian Sistem

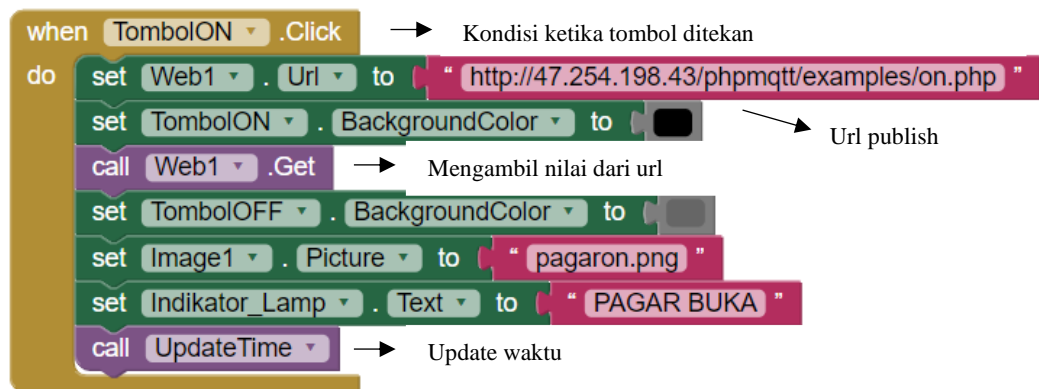
Pada tahap hasil pengujian sistem ini menjelaskan hasil bagaimana sistem tersebut dapat berjalan dengan lancar dan sesuai dengan perancangan sistem, perhitungan akurasi akan dilakukan menggunakan *Confussion Matrix*.

### 4.3.1 Kendali dengan *button*

Pada proses pengendalian dengan *button* ini, *Block code* pada *App Inventor* dapat dilihat pada Gambar 4.3 (a) dan (b) sebagai berikut.



(a)



(b)

**Gambar 4.3** (a) *Block code button off* (b) *Block code button on*

Pada Gambar 4.3 (a) merupakan *Block code* dari *button off* dimana ketika *button off* ditekan maka akan memanggil link pada gambar tersebut, link tersebut merupakan *code* dari PHPMQTT yang sudah tersimpan pada Alibaba Cloud ECS. Pada Gambar 4.3 (b) sama dengan Gambar 4.3 (a) *Block code* dari *button on* memanggil link sesuai pada gambar tersebut. Ketika setiap *button* ditekan maka akan mengirimkan pesan sesuai *button* yang ditekan dan bertindak sebagai *publish*, pesan kemudian di terima oleh *subscribe* dan akan menampilkan pesan, topic, dan waktu yang dikirim oleh *publish*, dapat dilihat pada Gambar 4.4 sebagai berikut.

```

root@user: /var/www/html/phpmqtt/examples
Sat, 17 Apr 2021 10:21:48 +0800: Fetching: 0 bytes
^C
root@user: /var/www/html/phpmqtt/examples# php subscribe.php
Sat, 17 Apr 2021 10:22:08 +0800: not had something in a while so ping
Sat, 17 Apr 2021 10:22:08 +0800: ping sent
Sat, 17 Apr 2021 10:22:08 +0800: Received CMD: 13 (PINGRESP)
Sat, 17 Apr 2021 10:22:08 +0800: Fetching: 0 bytes
Sat, 17 Apr 2021 10:22:09 +0800: Received CMD: 3 (PUBLISH)
Sat, 17 Apr 2021 10:22:09 +0800: Fetching: 38 bytes
Msg Recieved: Sat, 17 Apr 2021 10:22:09 +0800
Topic: bluerhinos/phpMQTT/examples/pagar

off
Sat, 17 Apr 2021 10:22:09 +0800: msg received but no match in subscriptions
Sat, 17 Apr 2021 10:22:15 +0800: Received CMD: 3 (PUBLISH)
Sat, 17 Apr 2021 10:22:15 +0800: Fetching: 37 bytes
Msg Recieved: Sat, 17 Apr 2021 10:22:15 +0800
Topic: bluerhinos/phpMQTT/examples/pagar

on
Sat, 17 Apr 2021 10:22:15 +0800: msg received but no match in subscriptions

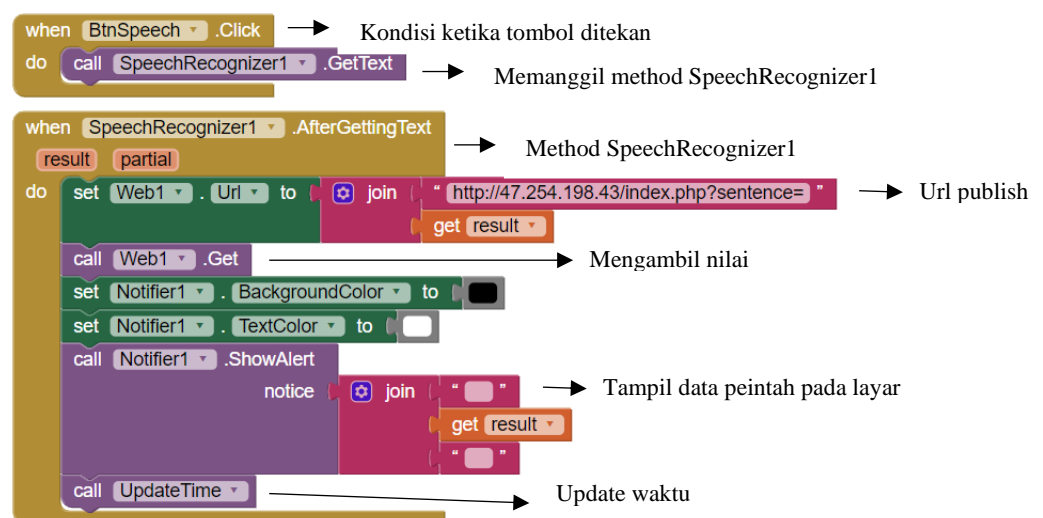
```

**Gambar 4.4** Pesan yang diterima oleh *subscribe*

Pada Gambar 4.4 tersebut hasil dari pesan yang dikirim oleh *publish* dan diterima oleh *subscribe* dengan *topic* “pagar”.

#### 4.3.2 Kendali dengan *Speech Recognition*

Pada proses dengan *Speech Recognition* ini, *Block code* pada *App Inventor* dapat dilihat pada Gambar 4.5 sebagai berikut.



**Gambar 4.5** *Block code Speech Recognition*

Pada Gambar 4.5 tersebut Ketika *button* ditekan dan mendapat perintah dalam Bahasa Indonesia akan memanggil link seperti pada gambar, dimana link tersebut akan menerima variabel “*sentence*” berupa *text* untuk diproses ke dalam algoritma *Boyer-Moore*.

#### 4.3.3 Proses *PreBmBc*

Proses *preBmBc* dilakukan agar mendapatkan nilai OH dimana nilai OH dinilai mulai dari *index* paling kanan pada *pattern*. Dalam proses *preBmBc* dilakukan pencacahan masing-masing karakter pada *pattern* dimulai dari *index* kanan sampai *index* paling kiri. Nilai pada setiap *index* diberikan sesuai rumus yang sudah peneliti jabarkan pada Bab 3. Adapaun hasil nilai OH pada masing-masing *pattern* dapat dilihat pada Gambar 4.6 sebagai berikut.

Pattern : buka pagar

PreBmBc : [9 8 7 6 5 4 3 2 1 ]

Pattern : tutup pagar

PreBmBc : [10 9 8 7 6 5 4 3 2 1 ]

Pattern : pagar buka

PreBmBc : [9 8 7 6 5 4 3 2 1 ]

Pattern : pagar tutup

PreBmBc : [10 9 8 7 6 5 4 3 2 1 ]

**Gambar 4.6** Nilai OH pada masing-masing *pattern*

Pada Gambar 4.6 tersebut merupakan hasil dari nilai OH pada masing-masing *pattern* yang sudah melalui proses pergeseran dan pencacahan.



#### 4.3.4 Proses *PreBmGs*

Proses *preBmGs* dilakukan agar mendapat nilai MH, agar mendapat nilai MH perlu dilakukan dengan membentuk *suffix* dari kanan ke kiri dari *pattern* dan perbandingan *suffix* kiri ke kanan dari *pattern*. Kemudian pemberian nilai MH dilakukan pencacahan dimulai dari *index* paling kanan *pattern* ke *index* paling kiri *pattern*, pada *index* paling kanan *pattern* selalu diberi nilai MH=1, selanjutnya dilakukan pengujian terhadap *suffix compare*. Apabila dalam proses pengujian terdapat kesamaan terhadap *suffix*, maka nilai MH karakter pada *pattern* sama dengan nilai pergeseran terhadap *suffix*, dan jika tidak terdapat kesamaan maka nilai MH karakter pada *pattern* bernilai sebanyak jumlah karakter ada *patteen*. Adapun hasil nilai MH dari proses *preBmGs* pada masing – masing *pattern* dapat dilihat pada Gambar 4.7 sebagai berikut.

Pattern : buka pagar

PreBmGs : [9 8 7 6 5 4 3 2 1 ]

Pattern : tutup pagar

PreBmGs : [10 9 8 7 6 5 4 3 2 1 ]

Pattern : pagar buka

PreBmGs : [9 8 7 6 5 4 3 2 1 ]

Pattern : pagar tutup

PreBmGs : [10 9 8 7 6 5 4 3 2 1 ]

**Gambar 4.7** Nilai MH pada masing-masing *pattern*

Pada Gambar 4.7 tersebut merupakan hasil dari nilai MH pada masing-masing *pattern* yang sudah melalui proses pergeseran dan pencacahan.

#### 4.3.5 Proses Pengujian *Boyer-Moore*

Pada proses pengujian *Boyer-Moore* ini dilakukan setelah didapatkan nilai OH dan MH pada proses sebelumnya. Algoritma *Boyer-Moore* melakukan pengujian *text* terhadap *pattern* dimulai dari karakter paling kanan pada *pattern*, kemudian dilakukan pergeseran sesuai dengan perbandingan *maximal* dari nilai OH pada karakter *text* dan MH pada karakter *pattern*. Apabila dalam proses pengujian benar maka akan melakukan proses pengiriman perintah *publish on* dan *publish off* dengan *topic* “pagar” ke *subscribe* yang berada pada MQTT di Alibaba Cloud ECS.

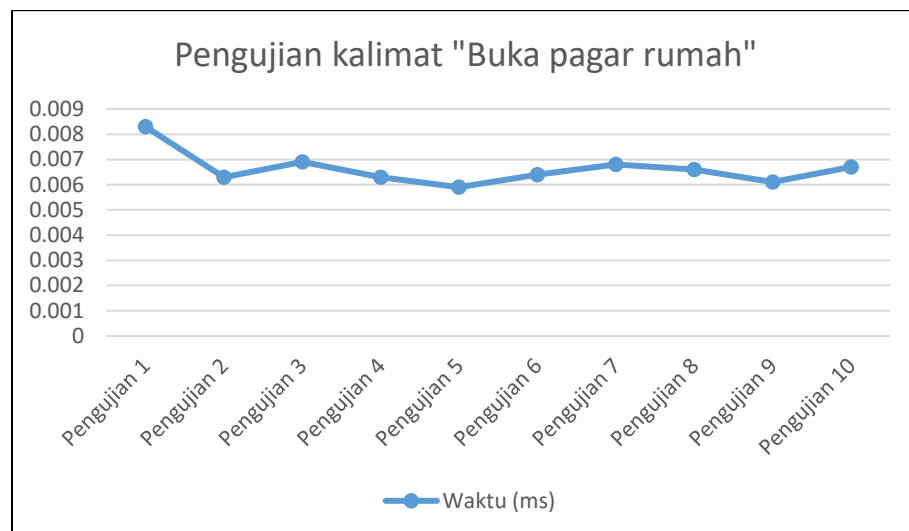
#### 4.3.6 Pengujian Algoritma *Boyer-Moore*

Algoritma *Boyer-Moore* pada sistem kendali pagar rumah ini diimplementasikan pada pengujian kalimat perintah *text* terhadap *pattern*, jumlah kalimat pengujian yang digunakan pada penelitian ini adalah 50 kalimat atau *text* dengan 4 *pattern*. Perhitungan akurasi pada sistem kendali pagar rumah dengan algoritma *Boyer-Moore* ini dilakukan dengan 10 kali pengujian pada setiap kalimat perintah menggunakan *Confussion Matrix* berdasarkan pada rumus 3.1. Pada Tabel 4.1 akan dilakukan pengujian pada kalimat perintah “Buka pagar rumah”, adapun hasil pengujian dapat dilihat pada Tabel 4.1 sebagai berikut.

**Tabel 4.1** Hasil pengujian kalimat “Buka pagar rumah”

Kalimat “Buka pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0083
2.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0063
3.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0069
4.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0063
5.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0059
6.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0064
7.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0068
8.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0066
9.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0061
10.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0067
Rata-rata					100%	0.00663

Grafik dari waktu pengujian pada kalimat “Buka pagar rumah” dapat dilihat pada Gambar 4.8 sebagai berikut.

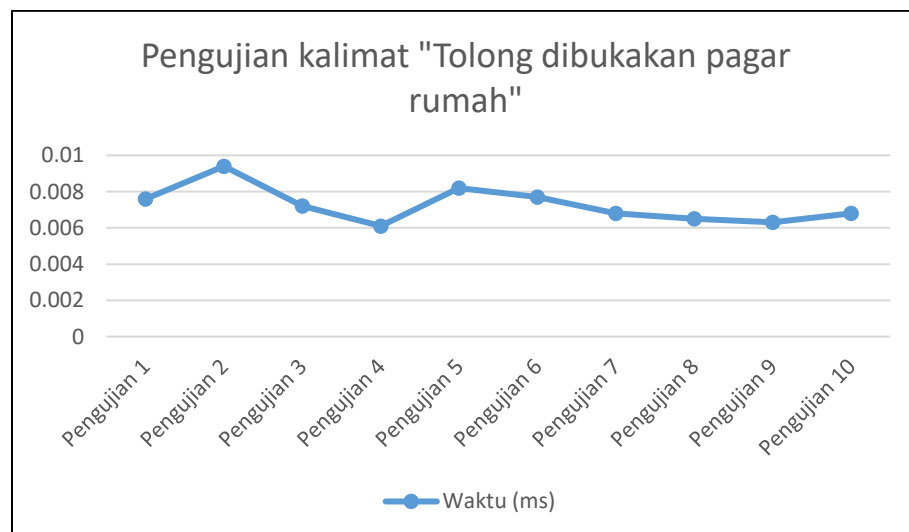
**Gambar 4.8** Grafik waktu pengujian kalimat “Buka pagar rumah”

Setelah dilakukan pengujian pada kalimat “Buka pagar rumah” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00663 ms. Selanjutnya dilakukan pengujian pada kalimat “Tolong dibukakan pagar rumah” memperoleh hasil sesuai pada Tabel 4.2 sebagai berikut.

**Tabel 4.2** Hasil pengujian kalimat “Tolong dibukakan pagar rumah”

Kalimat “Tolong dibukakan pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0076
2.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0094
3.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0072
4.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0061
5.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0082
6.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0077
7.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0068
8.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0065
9.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0063
10.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0068
Rata-rata					100%	0.00726

Grafik dari waktu pengujian pada kalimat “Tolong dibukakan pagar rumah” dapat dilihat pada Gambar 4.9 sebagai berikut.

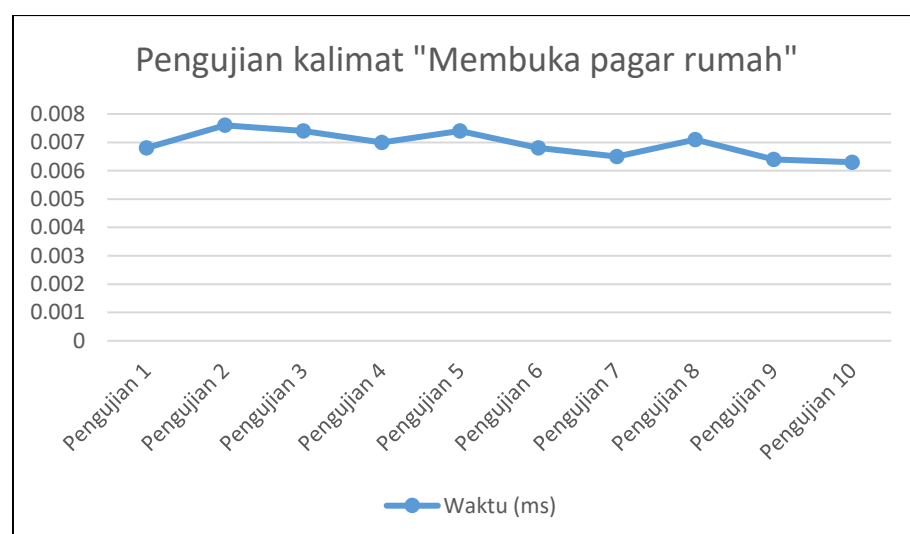
**Gambar 4.9** Grafik waktu pengujian kalimat “Tolong dibukakan pagar rumah”

Setelah dilakukan pengujian pada kalimat “Tolong dibukakan pagar rumah” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00726 ms. Selanjutnya dilakukan pengujian pada kalimat “Membuka pagar rumah” memperoleh hasil sesuai pada Tabel 4.3 sebagai berikut.

**Tabel 4.3** Hasil pengujian kalimat “Membuka pagar rumah”

Kalimat “Membuka pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0068
2.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0076
3.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0074
4.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.007
5.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0074
6.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0068
7.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0065
8.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0071
9.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0064
10.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0063
Rata-rata					100%	0.00693

Grafik dari waktu pengujian pada kalimat “Membuka pagar rumah” dapat dilihat pada Gambar 4.10 sebagai berikut.



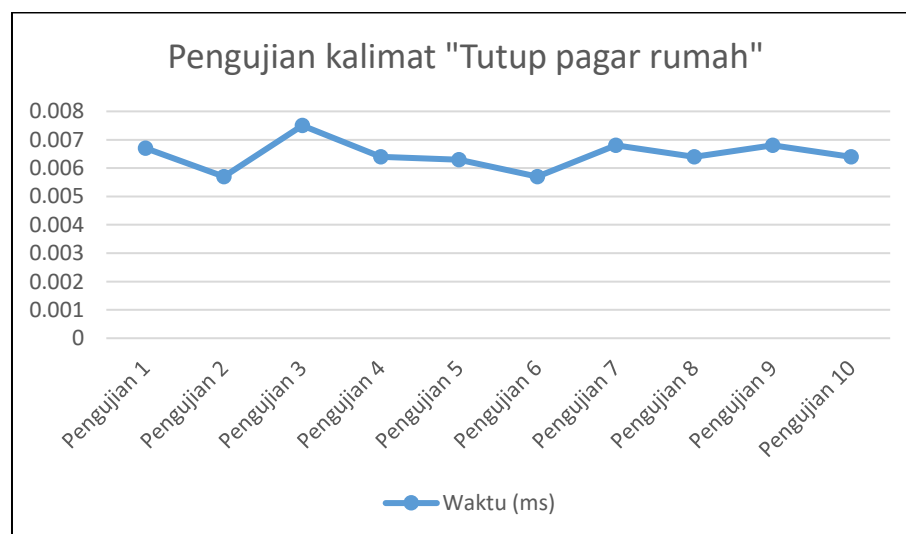
**Gambar 4.10** Grafik waktu pengujian kalimat “Membuka pagar rumah”

Setelah dilakukan pengujian pada kalimat “Membuka pagar rumah” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00693 ms. Selanjutnya dilakukan pengujian pada kalimat “Membuka pagar rumah” memperoleh hasil sesuai pada Tabel 4.4 sebagai berikut.

**Tabel 4.4** Hasil pengujian kalimat “Tutup pagar rumah”

Kalimat “Tutup pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0067
2.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0057
3.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0075
4.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0064
5.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0063
6.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0057
7.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0068
8.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0064
9.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0068
10.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0064
Rata-rata					100%	0.00647

Grafik dari waktu pengujian pada kalimat “Tutup pagar rumah” dapat dilihat pada Gambar 4.11 sebagai berikut.



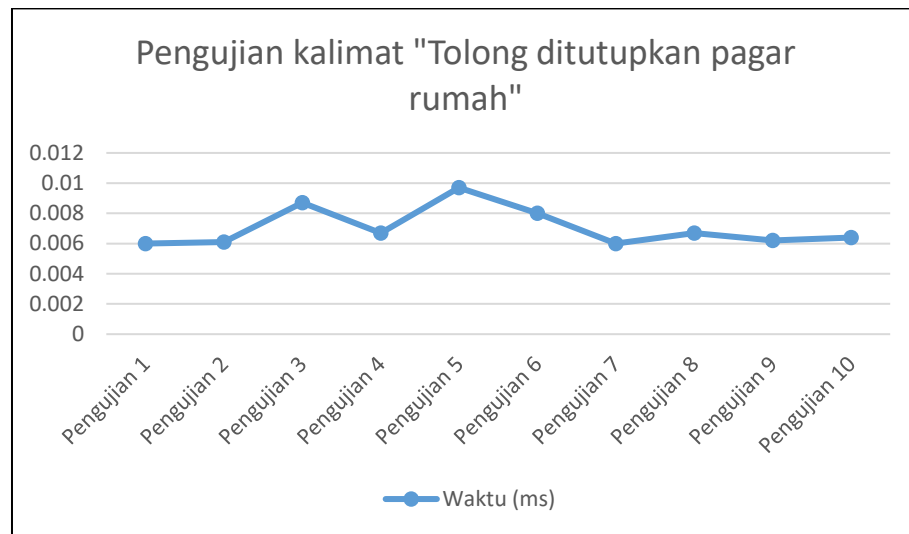
**Gambar 4.11** Grafik waktu pengujian kalimat “Tutup pagar rumah”

Setelah dilakukan pengujian pada kalimat “Tutup pagar rumah” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00647 ms. Selanjutnya dilakukan pengujian pada kalimat “Tolong ditutupkan pagar rumah” memperoleh hasil sesuai pada Tabel 4.5 sebagai berikut.

**Tabel 4.5** Hasil pengujian kalimat “Tolong ditutupkan pagar rumah”

Kalimat “Tolong ditutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.006
2.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0061
3.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0087
4.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0067
5.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0097
6.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.008
7.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.006
8.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0067
9.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
10.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0064
Rata-rata					100%	0.00705

Grafik dari waktu pengujian pada kalimat “Tolong ditutupkan pagar rumah” dapat dilihat pada Gambar 4.12 sebagai berikut.



**Gambar 4.12** Grafik waktu pengujian kalimat “Tolong ditutupkan pagar rumah”

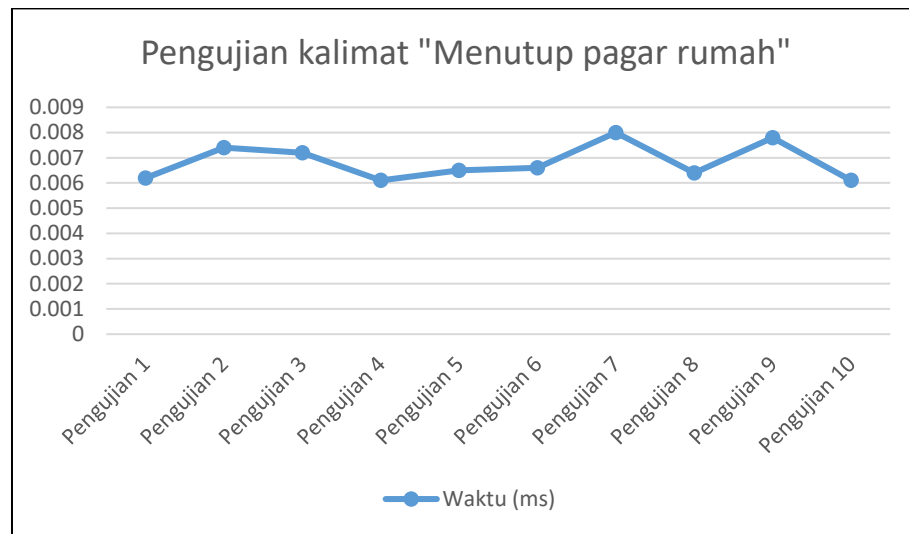
Setelah dilakukan pengujian pada kalimat “Tolong ditutupkan pagar rumah” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00705 ms. Selanjutnya dilakukan pengujian pada kalimat “Menutup pagar rumah” memperoleh hasil sesuai pada Tabel 4.6 sebagai berikut.

**Tabel 4.6** Hasil pengujian kalimat “Menutup pagar rumah”

Kalimat “Menutup pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
2.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0074
3.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0072
4.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0061
5.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0065
6.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0066
7.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.008
8.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0064
9.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0078
10.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0061
Rata-rata					100%	0.00683

Grafik dari waktu pengujian pada kalimat “Menutup pagar rumah” dapat dilihat pada Gambar 4.13 sebagai berikut.





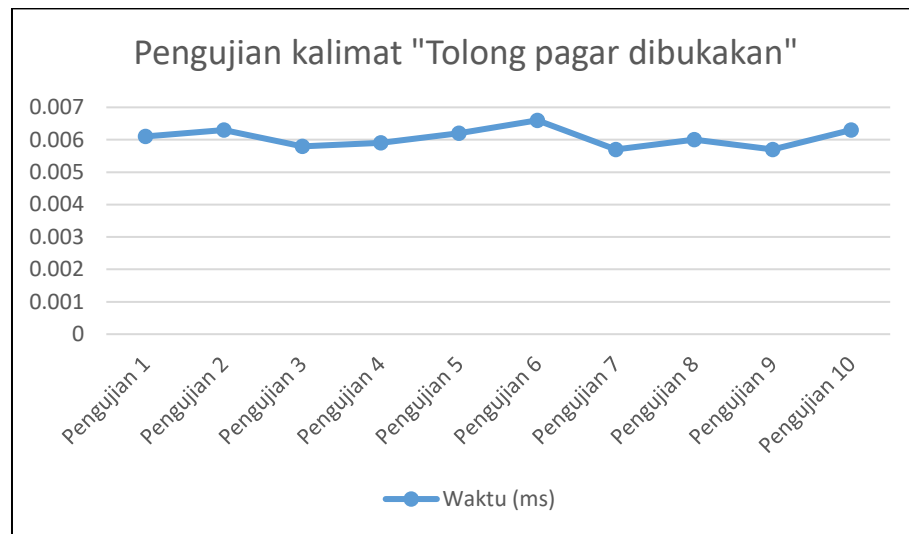
**Gambar 4.13** Grafik waktu pengujian kalimat “Menutup pagar rumah”

Setelah dilakukan pengujian pada kalimat “Menutup pagar rumah” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00683 ms. Selanjutnya dilakukan pengujian pada kalimat “Tolong pagar dibukakan “ memperoleh hasil sesuai pada Tabel 4.7 sebagai berikut.

**Tabel 4.7** Hasil pengujian kalimat “Tolong pagar dibukakan”

Kalimat “Tolong pagar dibukakan”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0061
2.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0063
3.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0058
4.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0059
5.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0062
6.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0066
7.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0057
8.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.006
9.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0057
10.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0063
Rata-rata					100%	0.00606

Grafik dari waktu pengujian pada kalimat “Tolong pagar dibukakan” dapat dilihat pada Gambar 4.14 sebagai berikut.



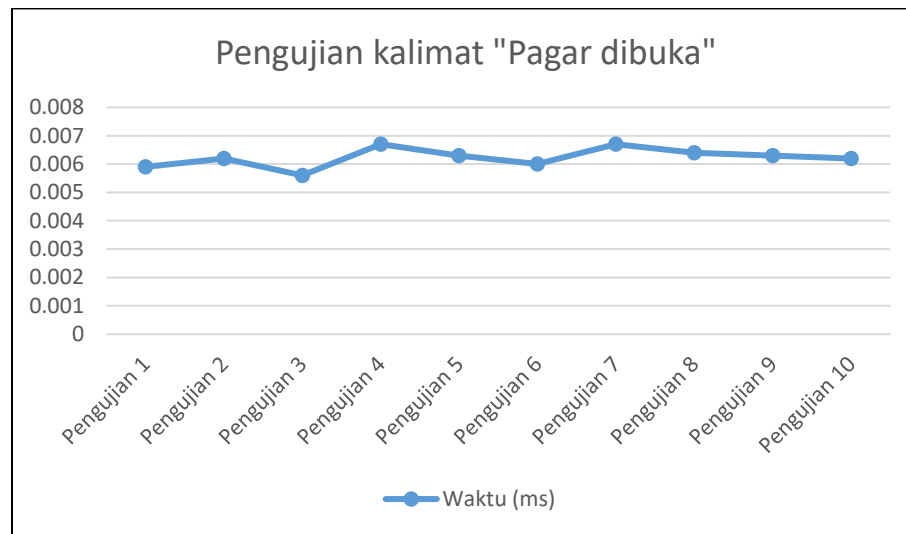
**Gambar 4.14** Grafik waktu pengujian kalimat “Tolong pagar dibukakan”

Setelah dilakukan pengujian pada kalimat “Tolong pagar dibukakan” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00606 ms. Selanjutnya dilakukan pengujian pada kalimat “Pagar dibuka” memperoleh hasil sesuai pada Tabel 4.8 sebagai berikut.

**Tabel 4.8** Hasil pengujian kalimat “Pagar dibuka”

Kalimat “Pagar dibuka”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0059
2.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0062
3.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0056
4.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0067
5.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0063
6.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.006
7.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0067
8.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0064
9.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0063
10.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0062
Rata-rata					100%	0.00623

Grafik dari waktu pengujian pada kalimat “Pagar dibuka” dapat dilihat pada Gambar 4.15 sebagai berikut.



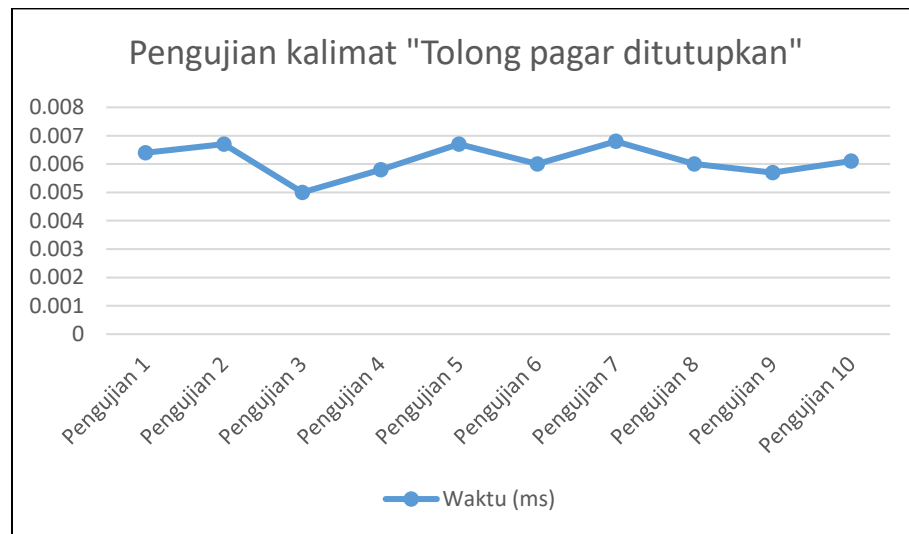
**Gambar 4.15** Grafik waktu pegujian kalimat “Pagar dibuka”

Setelah dilakukan pengujian pada kalimat “Pagar dibuka” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00623 ms. Selanjutnya dilakukan pengujian pada kalimat “Tolong pagar ditutupkan “ memperoleh hasil sesuai pada Tabel 4.9 sebagai berikut.

**Tabel 4.9** Hasil pegujian kalimat “Tolong pagar ditutupkan”

Kalimat “Tolong pagar ditutupkan”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0064
2.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0067
3.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.005
4.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0058
5.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0067
6.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.006
7.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0068
8.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.006
9.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0057
10.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0061
Rata-rata					100%	0.00612

Grafik dari waktu pengujian pada kalimat “Tolong pagar ditutupkan” dapat dilihat pada Gambar 4.16 sebagai berikut.



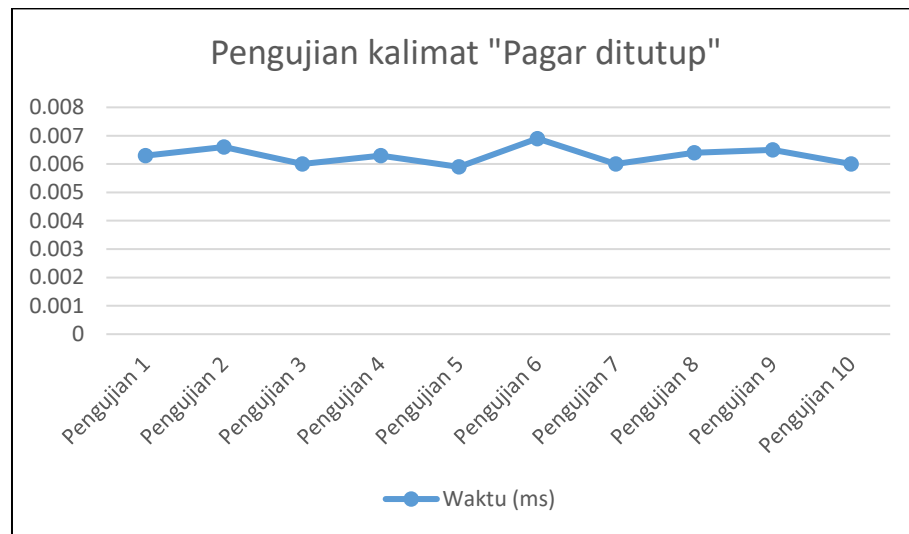
**Gambar 4.16** Grafik waktu pengujian kalimat “Tolong pagar ditutupkan”

Setelah dilakukan pengujian pada kalimat “Tolong pagar ditutup” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00612 ms. Selanjutnya dilakukan pengujian pada kalimat “Pagar ditutup” memperoleh hasil sesuai pada Tabel 4.10 sebagai berikut.

**Tabel 4.10** Hasil pengujian kalimat “Pagar ditutup”

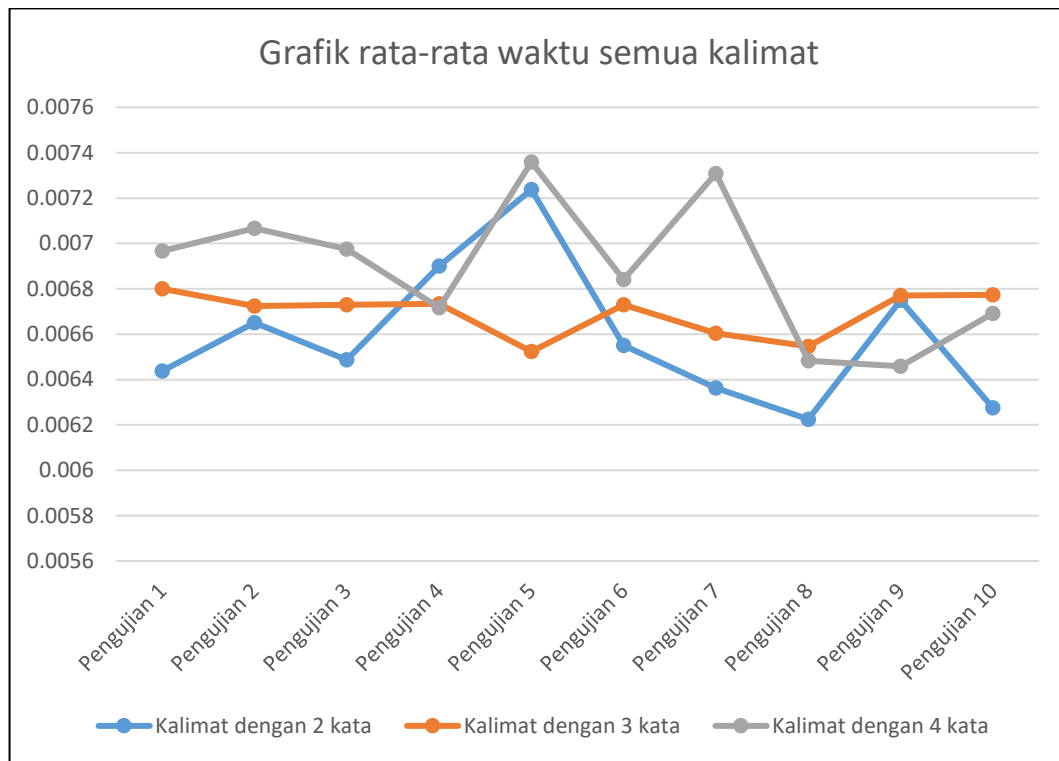
Kalimat “Pagar ditutup”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0063
2.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0066
3.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.006
4.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0063
5.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0059
6.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0069
7.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.006
8.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0064
9.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0065
10.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.006
Rata-rata					100%	0.00629

Grafik dari waktu pengujian pada kalimat “Tolong pagar ditutupkan” dapat dilihat pada Gambar 4.17 sebagai berikut.



**Gambar 4.17** Grafik waktu pegujian kalimat “Pagar ditutup”

Setelah dilakukan pengujian pada kalimat “Pagar ditutup” dengan 10 kali pengujian mendapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00629 ms. Hasil pengujian selanjutnya dapat dilihat pada bagian lampiran yang sudah peneliti cantumkan pada bagian akhir tulisan. Berikut hasil grafik waktu dari semua kalimat perintah yang terdiri dari 2 kata, 3 kata dan 4 kata dapat dilihat pada Gambar 4.18 sebagai berikut.



**Gambar 4.18** Grafik rata – rata waktu semua kalimat

Pada Gambar 4.18 tersebut merupakan grafik waktu dari pengujian semua kalimat yang terdiri dari 2 kata, 3 kata dan 4 kata, rata – rata yang dibutuhkan dalam pengujian untuk kalimat yang terdiri dari 2 kata adalah 0.00657 ms dan kalimat yang memiliki waktu terlama dalam 2 kata adalah pada kalimat ke 20 yaitu “Tutup pagar” pada pengujian 4 dan pengujian 5 dengan lama waktu 0.0099 ms, rata – rata yang dibutuhkan dalam pengujian untuk kalimat yang terdiri dari 3 kata adalah 0.00659 ms dan kalimat yang memiliki waktu terlama dalam 3 kata adalah pada kalimat 17 yaitu “Tolong membukakan pagar” pada pengujian 1 dan pengujian 9 dengan lama waktu masing – masing 0.01 ms dan 0.0103 ms, rata – rata yang dibutuhkan dalam pengujian untuk kalimat yang terdiri dari 4 kata adalah 0.00689 ms dan kalimat yang memiliki waktu terlama dalam 4 kata adalah

pada kalimat 16 yaitu “Tolong membuka pagar rumah” pada pengujian 1 dengan lama waktu 0.0104 ms. Pada Gambar 4.18 tersebut dilakukan 10 kali pengujian pada setiap kalimat dan memiliki selisih perubahan selisih waktu yang tidak sangat jauh. Dapat dilihat pada grafik garis warna abu – abu yaitu kalimat yang terdiri dari 4 kata memiliki waktu yang lebih lama dari kalimat yang terdiri dari 2 kata dan 3 kata. Lama waktu pengujian tersebut tergantung dari panjang *text* dari kalimat yang diuji, dapat dilihat pada rata – rata waktu dalam 2 kata yaitu 0.00657 ms sedangkan 3 kata yaitu 0.00659 ms dan 4 kata yaitu 00.00689 ms. Berdasarkan hasil uji coba dengan 10 kali pengujian pada 50 kalimat perintah dan hasil yang tepat dimasukkan ke dalam rumus akurasi, maka didapatkan hasil akurasi sebesar 100% dengan rata – rata waktu 0.00672 ms.

#### 4.4 Integrasi Islam

Sistem kendali pagar dengan perintah suara ini dibangun untuk kemudahan dan keamanan dalam membuka maupun menutup pagar rumah. Seperti dalam Al-Quran, Allah SWT menjelaskan tentang menjadikan rumah sebagai tempat tinggal, dalam surat *An-Nahl* ayat 80.

وَاللَّهُ جَعَلَ لَكُم مِّن بُيُوتِكُمْ سَكَنًا وَجَعَلَ لَكُم مِّن جُلُودِ الْأَنْعَامِ بُيُوتًا تَسْتَخِفُّونَهَا يَوْمَ  
ظَعْنِكُمْ وَيَوْمَ إِقَامَتِكُمْ<sup>٧</sup> وَمِنْ أَصْوَابِهَا وَأَوْبَارِهَا وَأَشْعَارِهَا أَثَاثًا وَمَتَاعًا إِلَىٰ حِينٍ

Artinya : *Dan Allah menjadikan bagimu rumah-rumahmu sebagai tempat tinggal dan Dia menjadikan bagi kamu rumah-rumah (kemah-kemah) dari kulit binatang ternak yang kamu merasa ringan (membawa)nya di waktu kamu berjalan dan waktu kamu bermukim dan (dijadikan-Nya pula) dari bulu domba, bulu unta dan bulu kambing, alat-alat rumah tangga dan perhiasan (yang kamu*

*pakai) sampai waktu (tertentu).* Pada lafal (وَاللّٰهُ جَعَلَ لَكُم مِّنْ بُيُوتِكُمْ سَكَنًا) tafsir Jalalain memiliki penafsiran rumah merupakan tempat kalian menetap didalamnya dan dalam tafsir lain Al-Madinah Al-Munawwarah dalam lafal tersebut memiliki penafsiran agar kalian tinggal di dalamnya dan diri kalian dapat beristirahat dari kegiatan.

Efisiensi waktu yang digunakan dalam pencarian sangat penting karena waktu yang dipakai tidak akan terbuang sia-sia saja karena satu proses yang seharusnya bisa dilakukan dengan cepat, seperti halnya pada agama islam, waktu adalah salah satu nikmat yang diberikan oleh Allah SWT kepada kita. Sudah pantasnya manusia memanfaatkannya dengan baik dan semaksimal mungkin. Islam menganjurkan agar manusia memanfaatkan waktu dan kesempatan yang dimiliki sehingga ia tidak termasuk ke dalam golongan orang-orang yang merugi. Hal ini dicantumkan dalam Qur'an Surat Al-Asr.

وَالْعَصْرِ (١) إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ (٢) إِلَّا الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَاصَوْا بِالْحَقِّ ۖ وَتَوَاصَوْا بِالصَّبْرِ (٣)

Artinya: “*Demi masa, sungguh, manusia berada dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan kebajikan serta saling menasihati untuk kebenaran dan saling menasihati untuk kesabaran.*”

Pada surat ini menunjukkan urgensi waktu. Surat ini berisi penegasan bahwa semua orang akan merugi kecuali orang-orang yang beriman dan beramal shalih serta saling menasehati agar menepati kebenaran dan kesabaran.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Penelitian yang telah dilakukan ini adalah penelitian tentang kendali pagar rumah dengan algoritma *Boyer-Moore*. Berdasarkan hasil uji pada penelitian ini sudah berjalan dengan baik, algoritma *Boyer-Moore* menghasilkan akurasi sebesar 100% dari 50 kalimat percobaan dengan 10 kali pengujian setiap kalimat perintah dan 4 *pattern* yang digunakan dengan durasi waktu pengujian keseluruhan kalimat dengan rata-rata 0.00672 ms. Lama waktu pengujian tersebut tergantung dari panjang *text* dari kalimat yang diuji, pada rata – rata waktu dalam kalimat yang terdiri dari 2 kata yaitu 0.00657 ms sedangkan 3 kata yaitu 0.00659 ms dan 4 kata yaitu 0.00689 ms.

#### 5.2. Saran

Sistem kendali pagar rumah dengan algoritma *Boyer-Moore* ini masih memiliki kekurangan, sehingga diperlukannya pengembangan sehingga diperoleh hasil yang lebih baik, berikut saran dari peneliti dapat bermanfaat bagi penelitian selanjutnya antara lain:

- a. Pengembangan lebih lanjut sistem tidak hanya dapat mengontrol pagar saja, tetapi kedepan dapat mengontrol banyak *device*.
- b. Kendali dengan *Speech Recognition* dapat dengan bermacam - macam Bahasa.

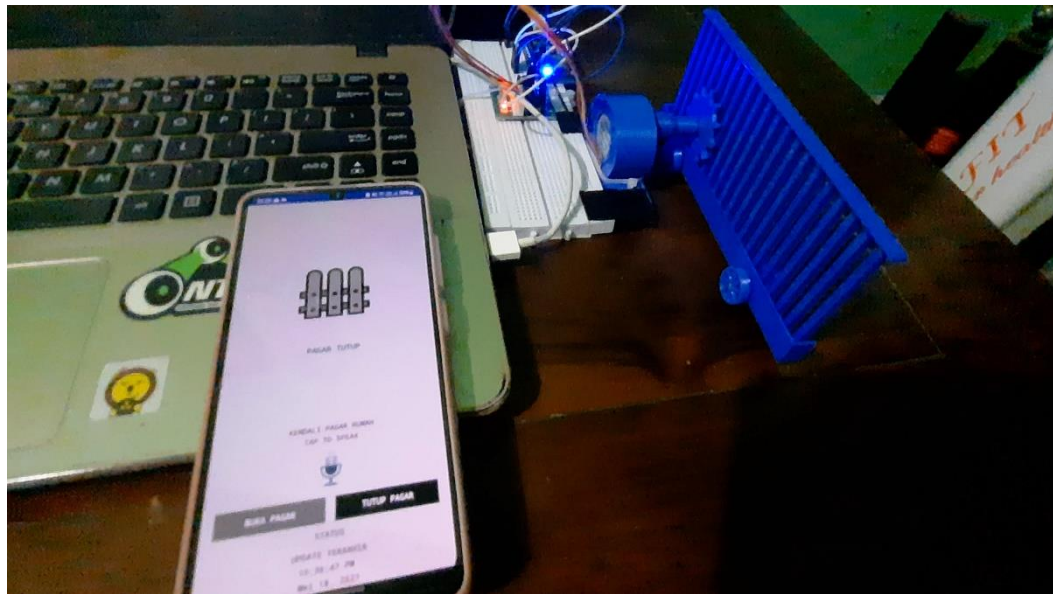
## DAFTAR PUSTAKA

- cyberthreat.id. (2020). *Digital 2020: Pengguna Internet Indonesia dalam Angka*.  
<https://cyberthreat.id/read/5387/Digital-2020-Pengguna-Internet-Indonesia-dalam-Angka>
- Darmawan, R. I., & Setianingrum, A. H. (2018). Implementasi Algoritma Boyer Moore Pada Aplikasi Kamus Istilah Kebidanan Berbasis Web. *Jurnal Sistem Informasi*, 02(01), 10.
- Dirmansyah, J., & Hermanto, D. (2014). Rancang Bangun Aplikasi Penunjuk Arah Berbahasa Indonesia Berbasis Text To Speech Dan Speech Recognition Pada Perangkat Android. *Fakultas Teknologi Informasi Dan Elektro Universitas Teknologi Yogyakarta*, 8.
- Effendi, D., Hartono, T., & Kurnaedi, A. (2013). Penerapan String Matching Menggunakan Algoritma Boyer-moore Pada Translator Bahasa Pascal Ke C. *Majalah Ilmiah UNIKOM*, Vol.11 No. 2, 14.
- Junaidi, B. D. (2018). Implementasi Pattern Matching pada Speech Recognition. *Institut Teknologi Bandung, Makalah IF2211 Strategi Algoritma*, 9.
- kominfo.go.id. (2015). *Indonesia Raksasa Teknologi Digital Asia*.  
<https://kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia>
- Kumara, G. H. (2008). Visualisasi Beberapa Algoritma Pencocokan String Dengan Java. *Institut Teknologi Bandung*, 14.
- Nursyahrina. (2016). Aplikasi Pencocokan Pola pada Asisten Pribadi Cerdas Google Now. *Makalah IF2211 Strategi Algoritma*.

- Orwant, J., Hietaniemi, J., & Macdonald, J. (1999). *Mastering Algorithms R'ith Perl*. O'Reilly United States.
- Prakasa, O. S. Y., & Lhaksamana, K. M. (2018). Klasifikasi Teks Dengan Menggunakan Algoritma K-nearest Neighbor Pada Kasus Kinerja Pemerintah Di Twitter. *e-Proceeding of Engineering, Vol.5, No.3*, 12.
- Rani, M. P. J., Bakthakumar, J., & Kumar, S. (2017). Voice Controlled Home Automation System Using Natural Language Processing (Nlp) And Internet of Things (IoT). *International Conference on Science Technology Engineering & Management (ICONSTEM)*, 6.
- Surat Al-Baqarah Ayat 125. (2020). Tafsir AlQuran Online. <https://tafsirq.com/2-al-baqarah/ayat-125>
- Utama, H. S., Setiawan, J., & Mardjoko, P. B. (2020). Sistem Kontrol Pintu Pagar Rumah Berbasis Arduino dengan Koneksi Nirkabel Bluetooth pada Smartphone Android. *TESLA: Jurnal Teknik Elektro*, 21(2), 135. <https://doi.org/10.24912/tesla.v21i2.7184>
- Wibowo, J. (2016). Aplikasi Penentuan Kata Dasar Dari Kata Berimbuhan Pada Kalimat Bahasa Indonesia Dengan Algoritma Stemming. *Jurnal Riset Komputer (JURIKOM)*, Volume : 3, Nomor: 5, 5.

## LAMPIRAN

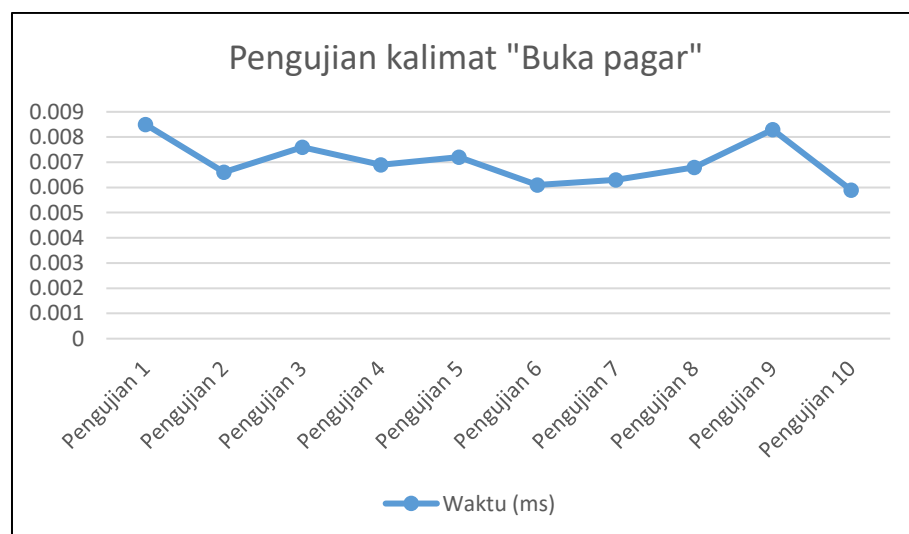
Link video demo project “Sistem kendali pagar rumah berbasis *Internet of Things* dengan perintah suara menggunakan algoritma *Boyer-Moore*”.



Link : <http://bit.ly/demo17650057>

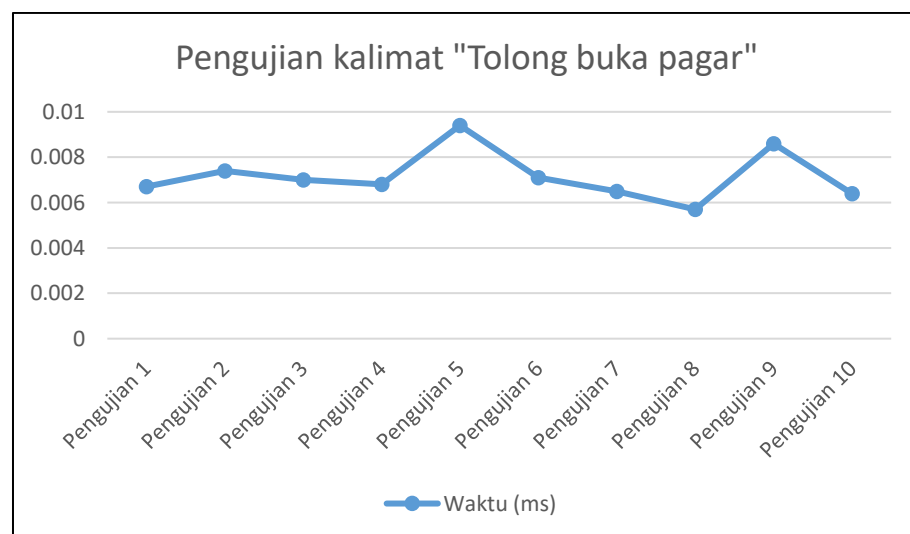
Hasil pengujian kalimat “Buka pagar”.

Kalimat “Buka pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0085
2.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0066
3.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0076
4.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0069
5.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0072
6.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0061
7.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0063
8.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0068
9.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0083
10.	Buka pagar	Buka pagar	0 Karakter	Tepat	100%	0.0059
Rata-rata					100%	0.00702



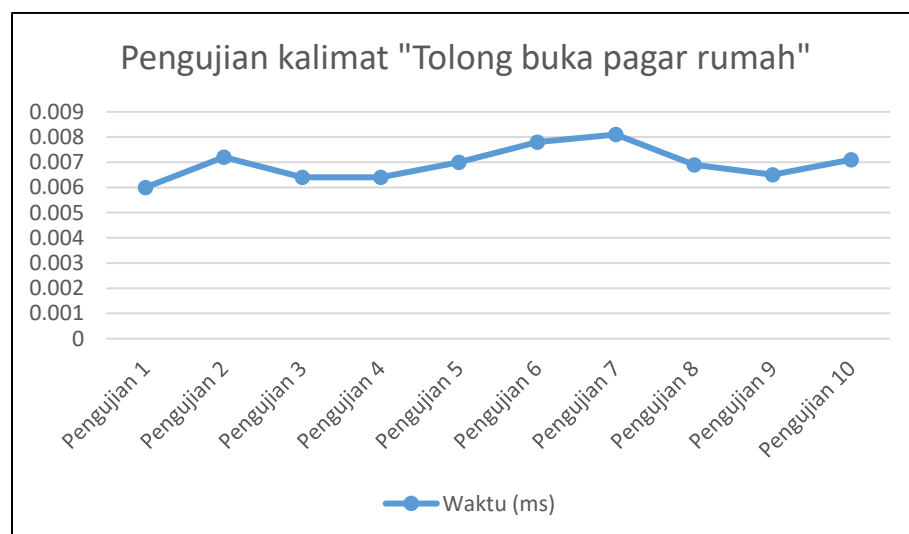
Hasil pengujian kalimat “Tolong buka pagar”.

Kalimat “Tolong buka pagar”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0067
2.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0074
3.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.007
4.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0068
5.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0094
6.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0071
7.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0065
8.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0057
9.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0086
10.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0064
Rata-rata					100%	0.00716



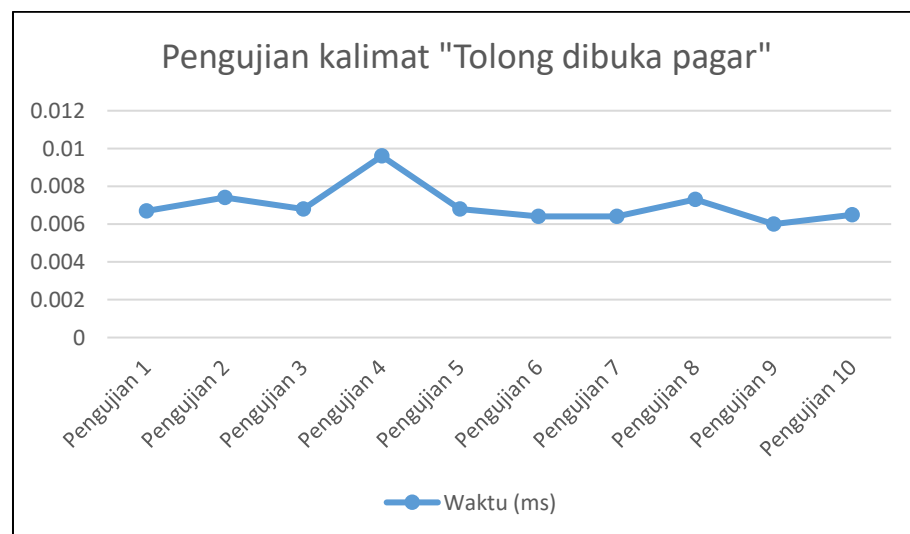
Hasil pengujian kalimat “Tolong buka pagar rumah”.

Kalimat “Tolong buka pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.006
2.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0072
3.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0064
4.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0064
5.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.007
6.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0078
7.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0081
8.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0069
9.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0065
10.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0071
Rata-rata					100%	0.00694



Hasil pengujian kalimat “Tolong dibuka pagar”.

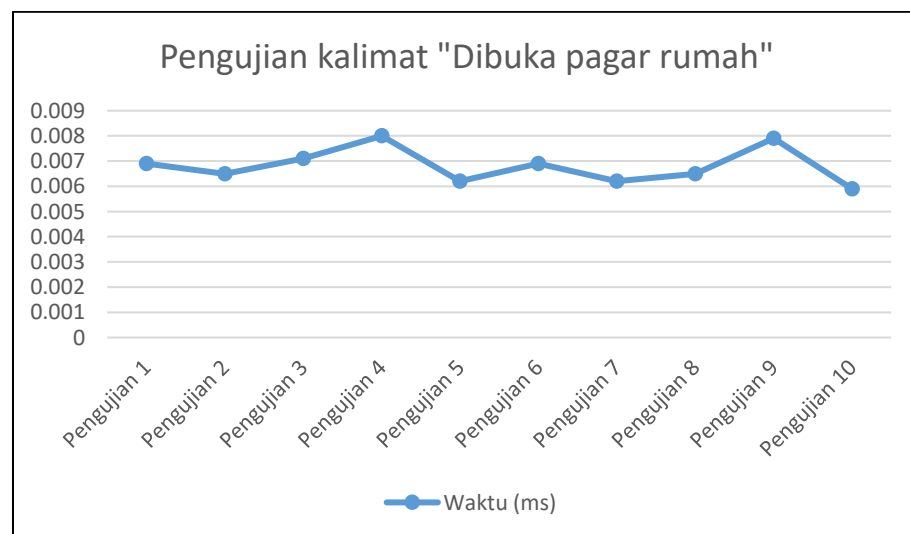
Kalimat “Tolong dibuka pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0067
2.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0074
3.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0068
4.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0096
5.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0068
6.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0064
7.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0064
8.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0073
9.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.006
10.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0065
Rata-rata					100%	0.00699





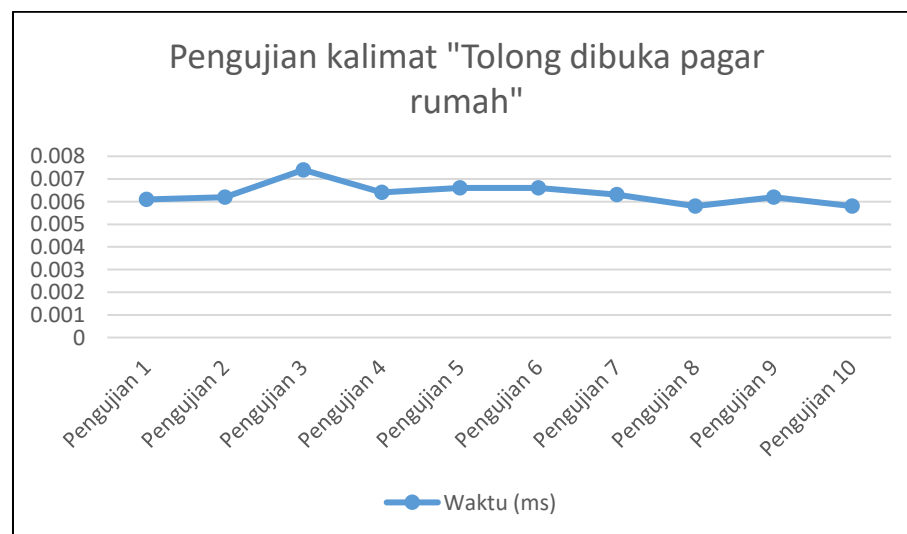
Hasil pengujian kalimat “Dibuka pagar rumah”.

Kalimat “Dibuka pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0069
2.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0065
3.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0071
4.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.008
5.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0062
6.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0069
7.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0062
8.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0065
9.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0079
10.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0059
Rata-rata					100%	0.00681



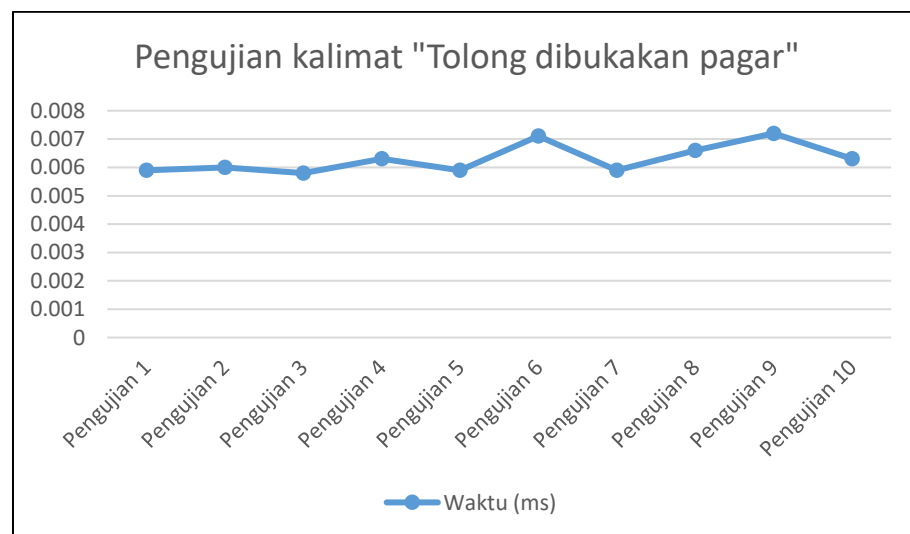
Hasil pengujian kalimat “Tolong dibuka pagar rumah”.

Kalimat “Tolong dibuka pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0061
2.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0062
3.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0074
4.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0064
5.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0066
6.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0066
7.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0063
8.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0058
9.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0062
10.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0058
Rata-rata					100%	0.00634



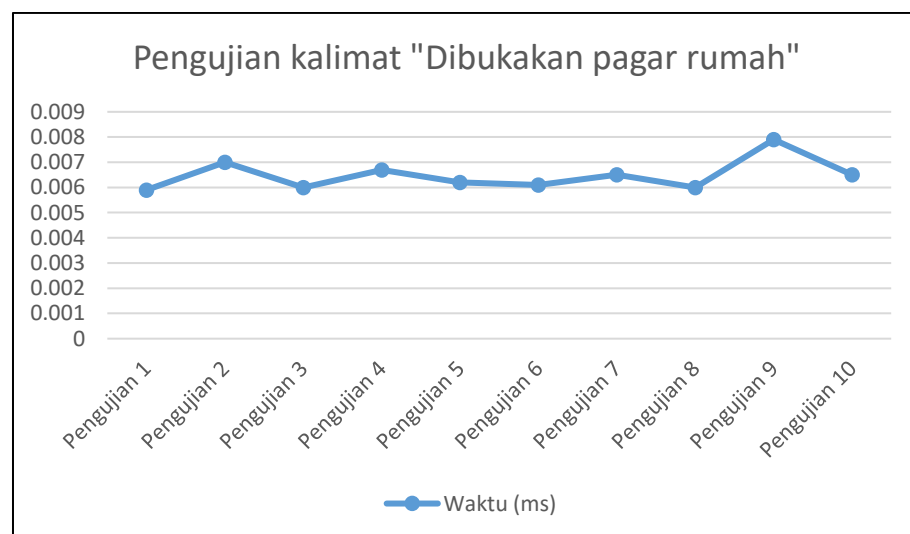
Hasil pengujian kalimat “Tolong dibukakan pagar”.

Kalimat “Tolong dibukakan pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0059
2.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.006
3.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0058
4.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0063
5.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0059
6.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0071
7.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0059
8.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0066
9.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0072
10.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0063
Rata-rata					100%	0.0063



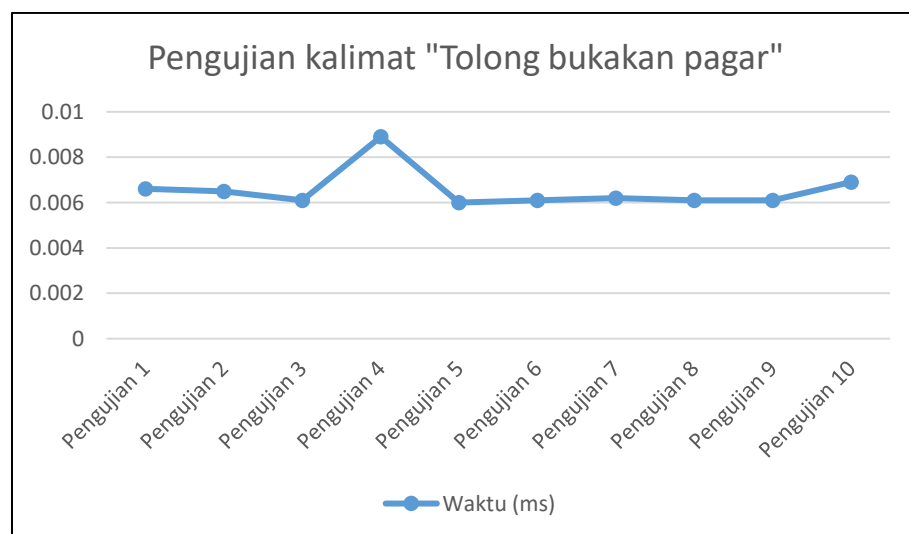
Hasil pengujian kalimat “Dibukakan pagar rumah”.

Kalimat “Dibukakan pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0059
2.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.007
3.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.006
4.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0067
5.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0062
6.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0061
7.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0065
8.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.006
9.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0079
10.	Bukakan pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0065
Rata-rata					100%	0.00648



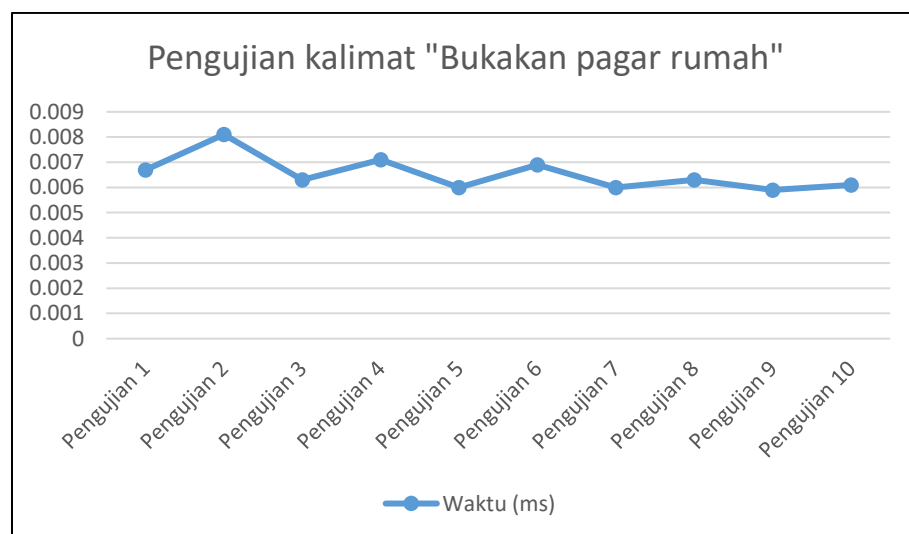
Hasil pengujian kalimat “Tolong bukakan pagar”.

Kalimat “Tolong bukakan pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0066
2.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0065
3.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0061
4.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0089
5.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.006
6.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0061
7.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0062
8.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0061
9.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0061
10.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0069
Rata-rata					100%	0.00655



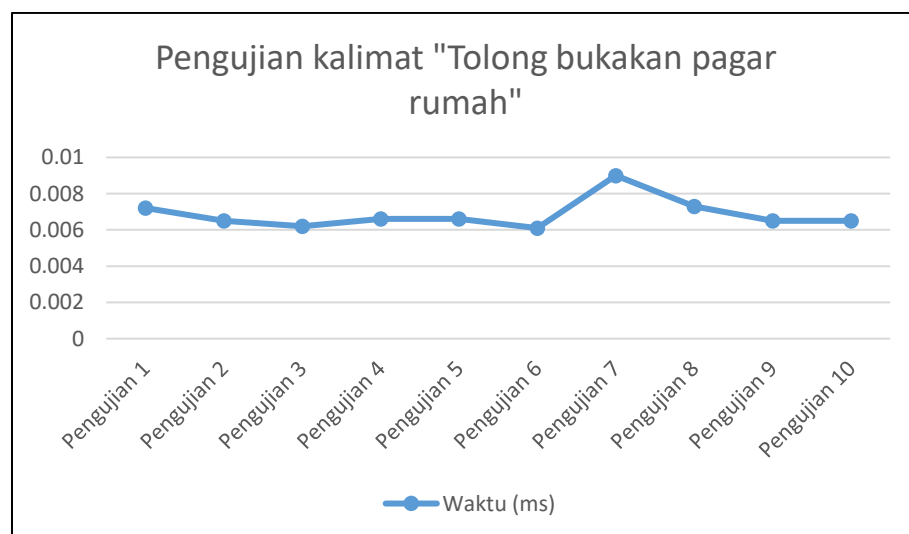
Hasil pengujian kalimat “Bukakan pagar rumah”.

Kalimat “Bukakan pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0067
2.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0081
3.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0063
4.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0071
5.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.006
6.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0069
7.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.006
8.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0063
9.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0059
10.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0061
Rata-rata					100%	0.00654



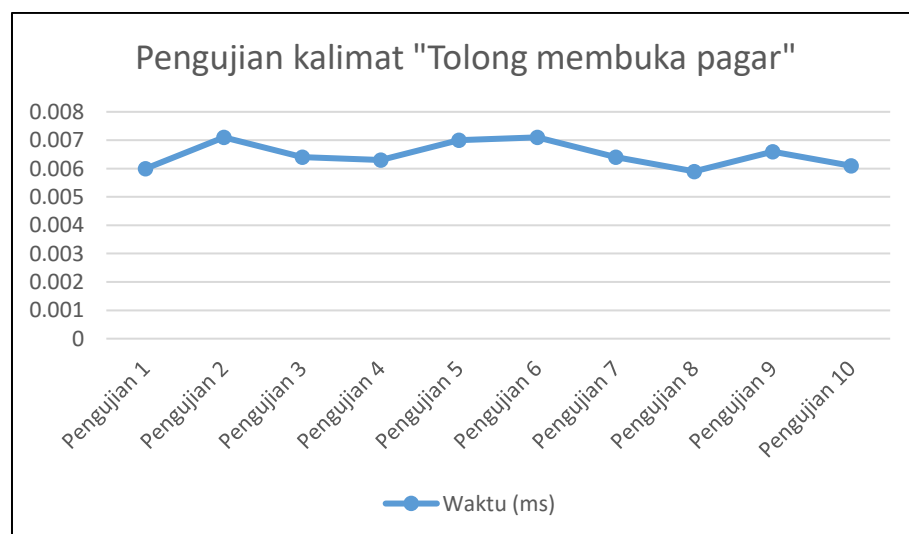
Hasil pengujian kalimat “Tolong bukakan pagar rumah”.

Kalimat “Tolong bukakan pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0072
2.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0065
3.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0062
4.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0066
5.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0066
6.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0061
7.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.009
8.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0073
9.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0065
10.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0065
Rata-rata					100%	0.00685



Hasil pengujian kalimat “Tolong membuka pagar”.

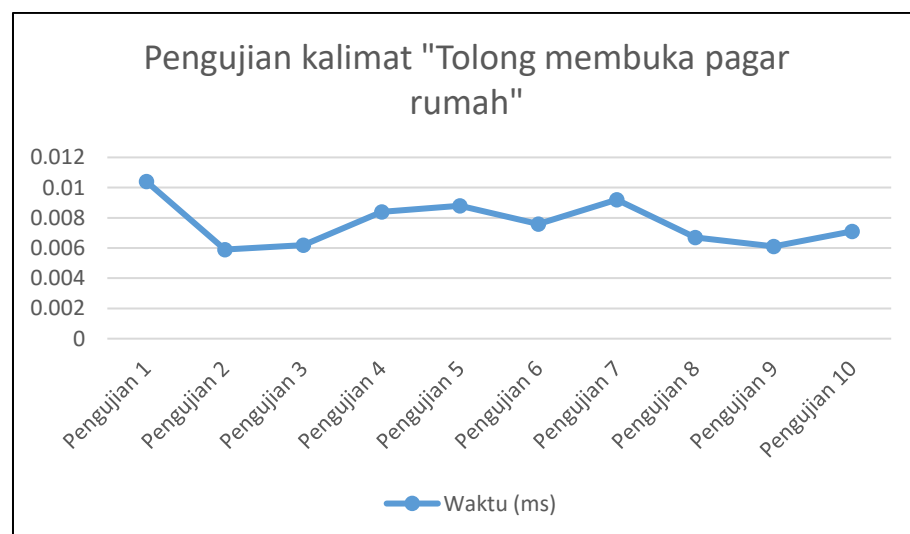
Kalimat “Tolong membuka pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.006
2.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0071
3.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0064
4.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0063
5.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.007
6.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0071
7.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0064
8.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0059
9.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0066
10.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0061
Rata-rata					100%	0.00649





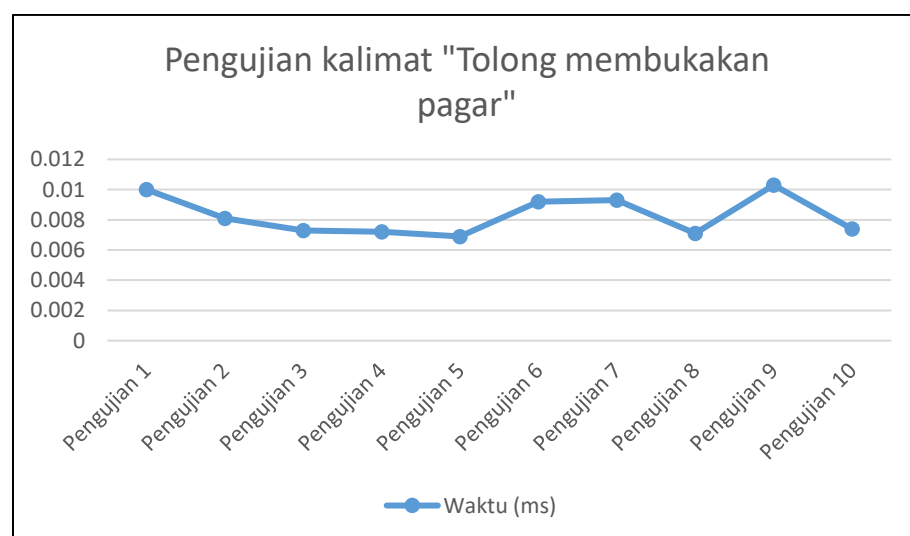
Hasil pengujian kalimat “Tolong membuka pagar rumah”.

Kalimat “Tolong membuka pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0104
2.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0059
3.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0062
4.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0084
5.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0088
6.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0076
7.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0092
8.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0067
9.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0061
10.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0071
Rata-rata					100%	0.00764



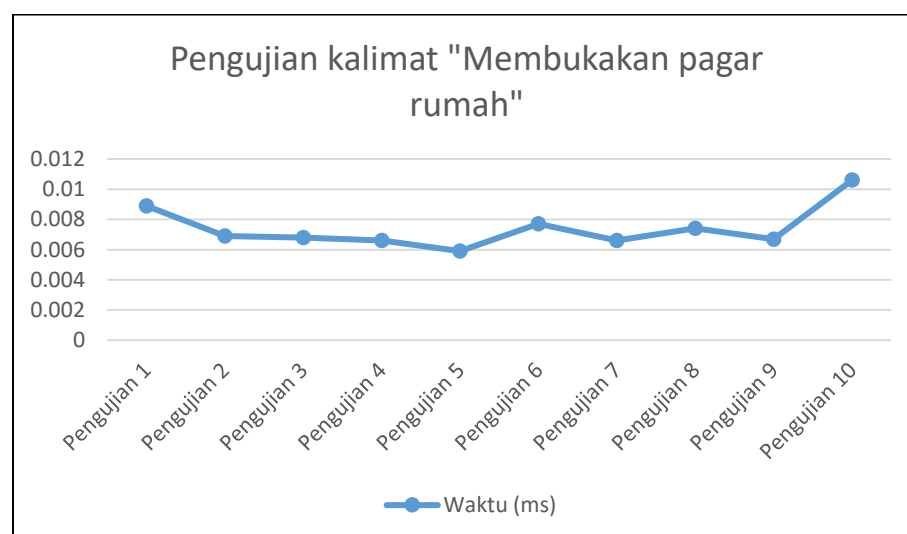
Hasil pengujian kalimat “Tolong membukakan pagar”.

Kalimat “Tolong membukakan pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.01
2.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0081
3.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0073
4.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0072
5.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0069
6.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0092
7.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0093
8.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0071
9.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0103
10.	Tolong buka pagar	Buka pagar	7 Karakter	Tepat	100%	0.0074
Rata-rata					100%	0.00828



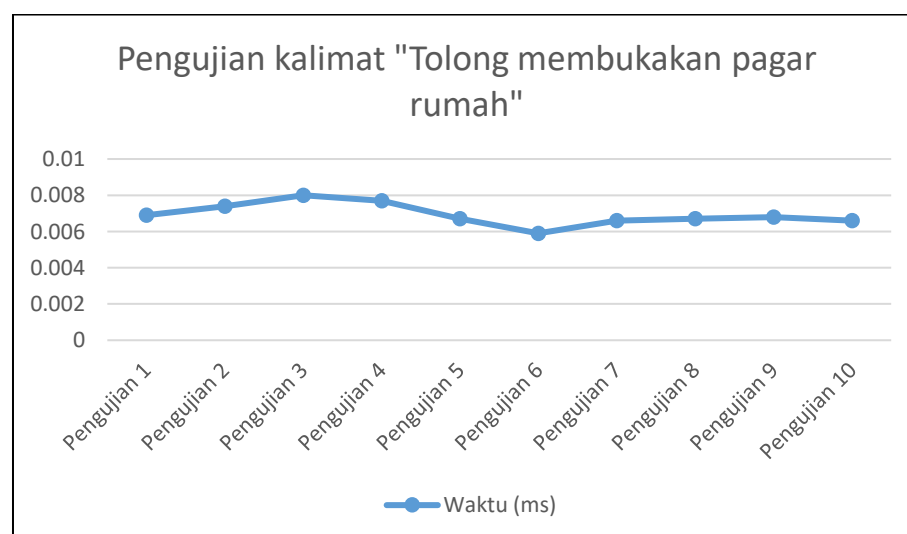
Hasil pengujian kalimat “Membukakan pagar rumah”.

Kalimat “Membukakan pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0089
2.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0069
3.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0068
4.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0066
5.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0059
6.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0077
7.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0066
8.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0074
9.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0067
10.	Buka pagar rumah	Buka pagar	0 Karakter	Tepat	100%	0.0106
Rata-rata					100%	0.00741



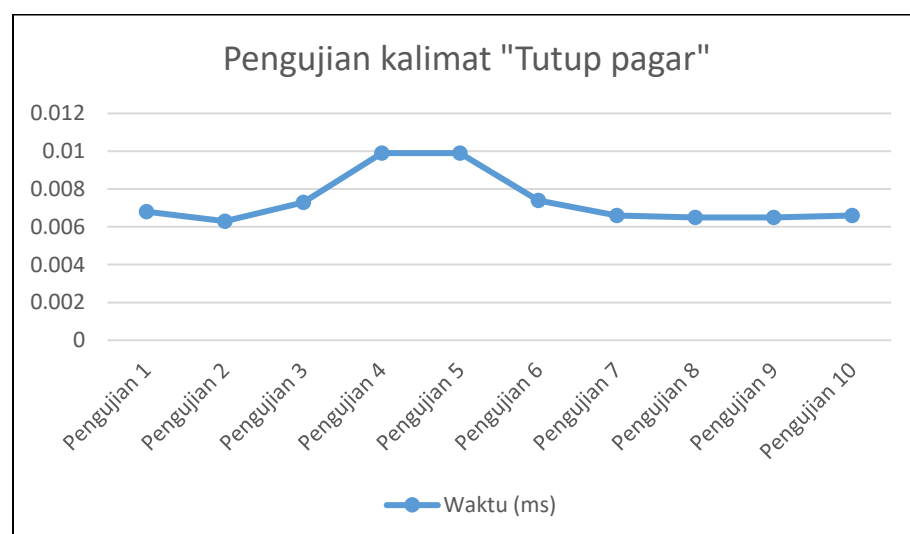
Hasil pengujian kalimat “Tolong membukakan pagar rumah”.

Kalimat “Tolong membukakan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0069
2.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0074
3.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.008
4.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0077
5.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0067
6.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0059
7.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0066
8.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0067
9.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0068
10.	Tolong buka pagar rumah	Buka pagar	7 Karakter	Tepat	100%	0.0066
Rata-rata					100%	0.00693



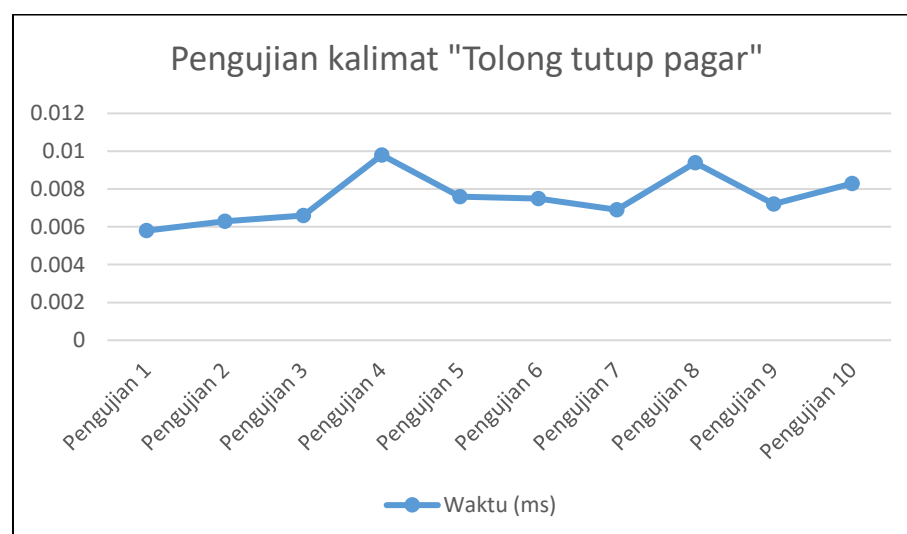
Hasil pengujian kalimat “Tutup pagar”.

Kalimat “Tutup pagar”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0068
2.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0063
3.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0073
4.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0099
5.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0099
6.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0074
7.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0066
8.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0065
9.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0065
10.	Tutup pagar	Tutup pagar	0 Karakter	Tepat	100%	0.0066
Rata-rata					100%	0.00738



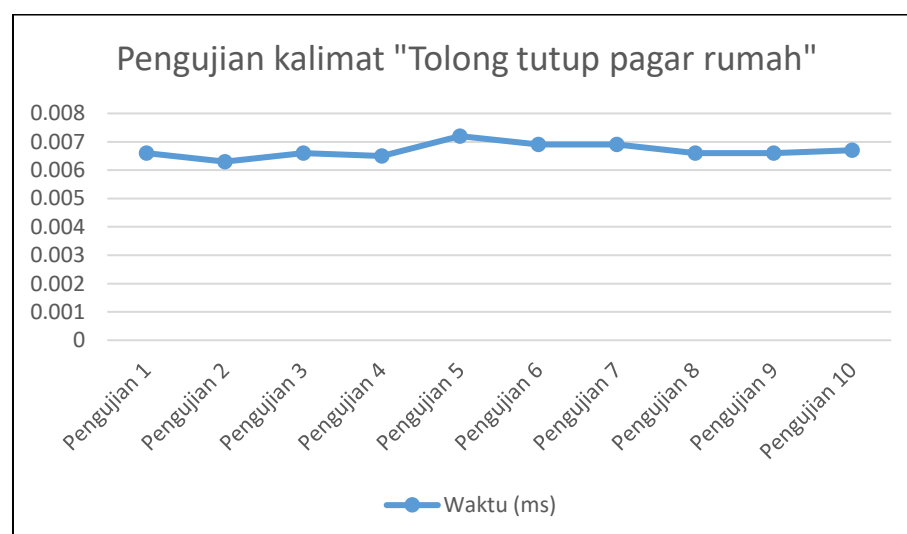
Hasil pengujian kalimat “Tolong tutup pagar”.

Kalimat “Tutup pagar”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0058
2.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
3.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0066
4.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0098
5.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0076
6.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0075
7.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0069
8.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0094
9.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0072
10.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0083
Rata-rata					100%	0.00754



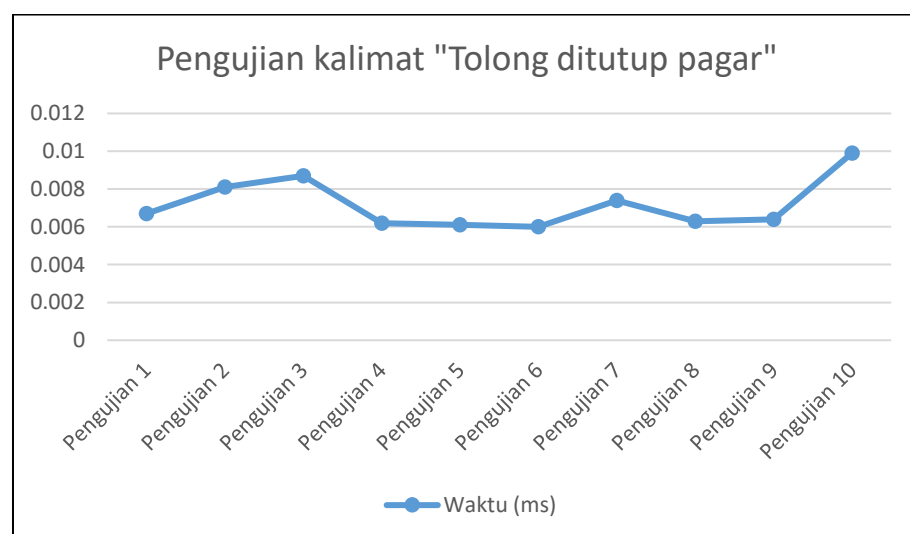
Hasil pengujian kalimat “Tolong tutup pagar rumah”.

Kalimat “Tolong tutup pagar rumah”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0066
2.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0063
3.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0066
4.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0065
5.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0072
6.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0069
7.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0069
8.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0066
9.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0066
10.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0067
Rata-rata					100%	0.00669



Hasil pengujian kalimat “Tolong ditutup pagar”.

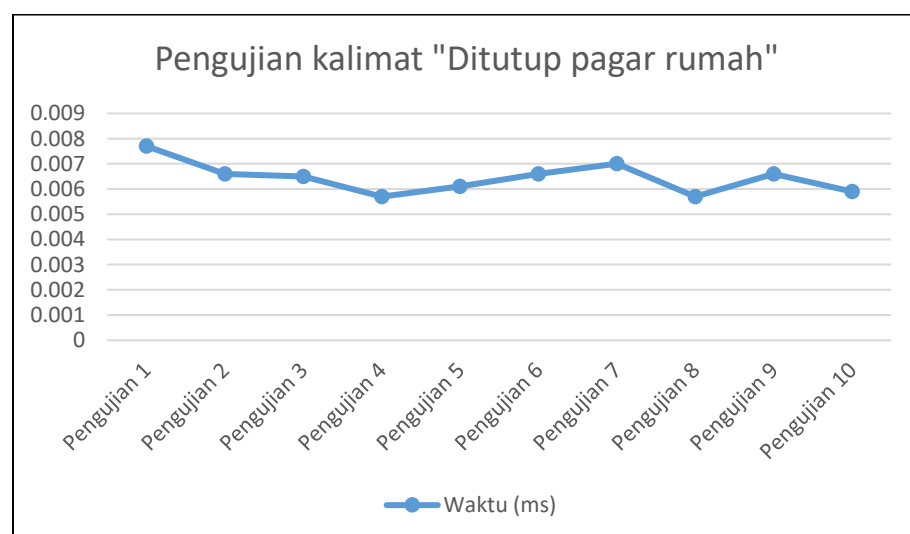
Kalimat “Tolong ditutup pagar”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0067
2.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0081
3.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0087
4.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0062
5.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0061
6.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.006
7.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0074
8.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
9.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0064
10.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0099
Rata-rata					100%	0.00718





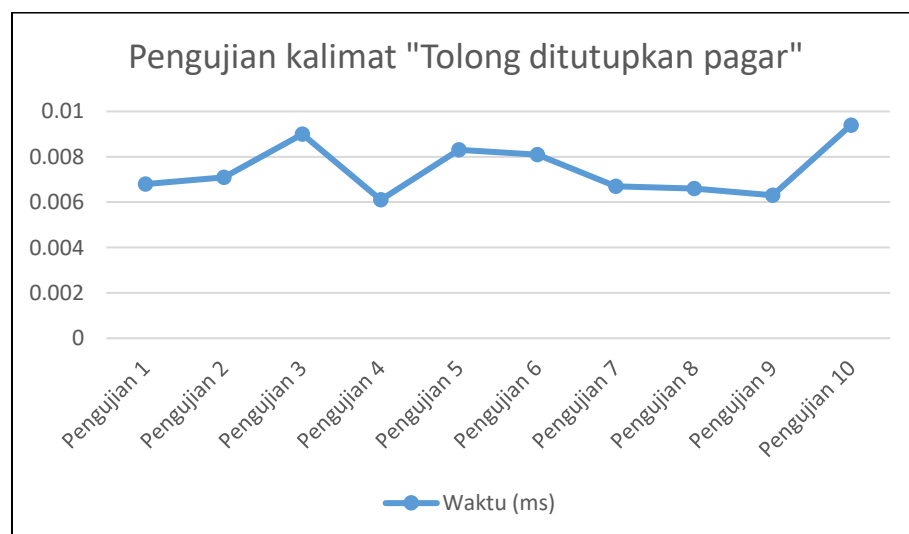
Hasil pengujian kalimat “Ditutup pagar rumah”.

Kalimat “Ditutup pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0077
2.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0066
3.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0065
4.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0057
5.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0061
6.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0066
7.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.007
8.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0057
9.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0066
10.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0059
Rata-rata					100%	0.00644



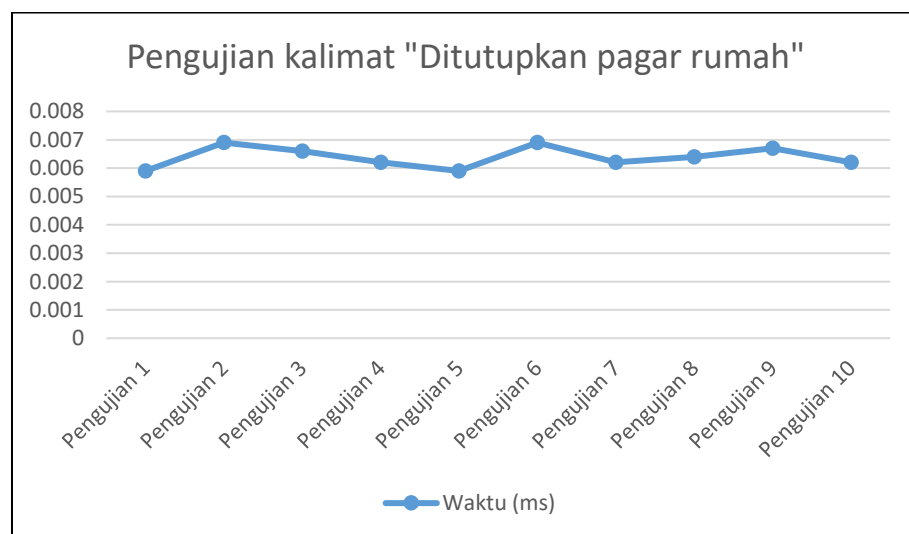
Hasil pengujian kalimat “Tolong ditutupkan pagar”.

Kalimat “Tolong ditutupkan pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0068
2.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0071
3.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.009
4.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0061
5.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0083
6.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0081
7.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0067
8.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0066
9.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
10.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0094
Rata-rata					100%	0.00744



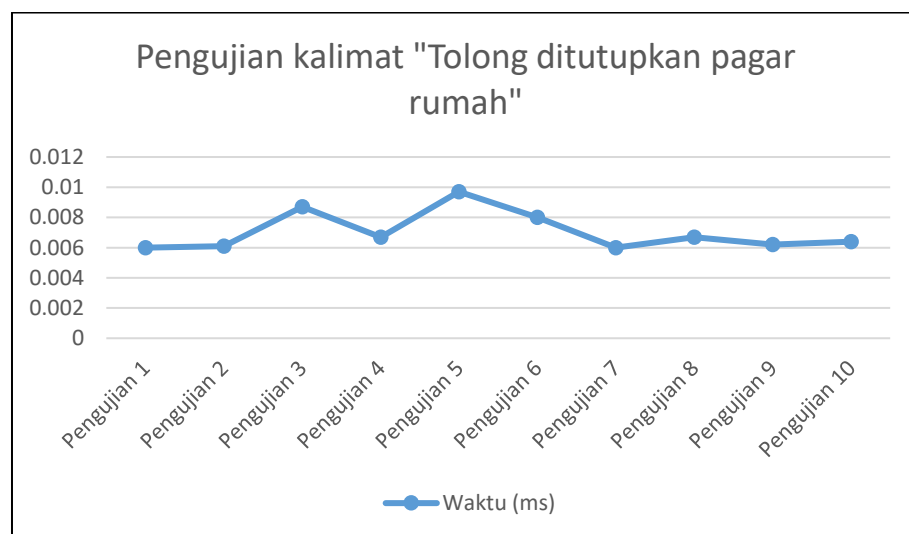
Hasil pengujian kalimat “Ditutupkan pagar rumah”.

Kalimat “Ditutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0059
2.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0069
3.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0066
4.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
5.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0059
6.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0069
7.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
8.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0064
9.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0067
10.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
Rata-rata					100%	0.00639



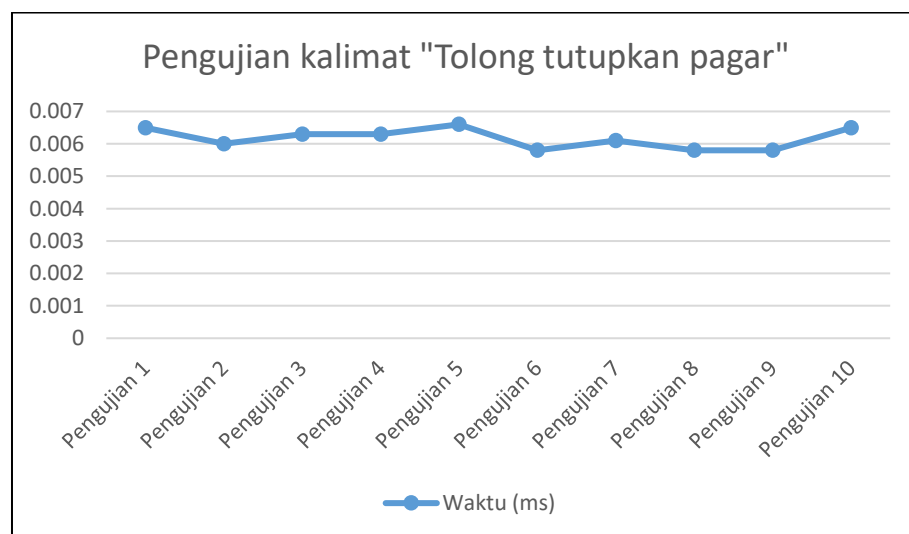
Hasil pengujian kalimat “Tolong ditutupkan pagar rumah”.

Kalimat “Tolong ditutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.006
2.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0061
3.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0087
4.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0067
5.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0097
6.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.008
7.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.006
8.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0067
9.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
10.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0064
Rata-rata					100%	0.00705



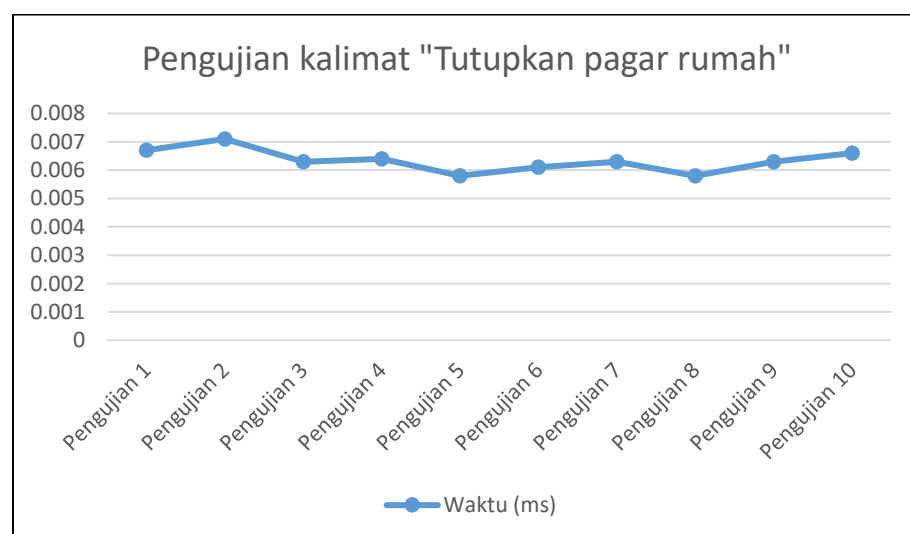
Hasil pengujian kalimat “Tolong tutupkan pagar”.

Kalimat “Tolong tutupkan pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0065
2.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.006
3.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
4.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
5.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0066
6.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0058
7.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0061
8.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0058
9.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0058
10.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0065
Rata-rata					100%	0.00617



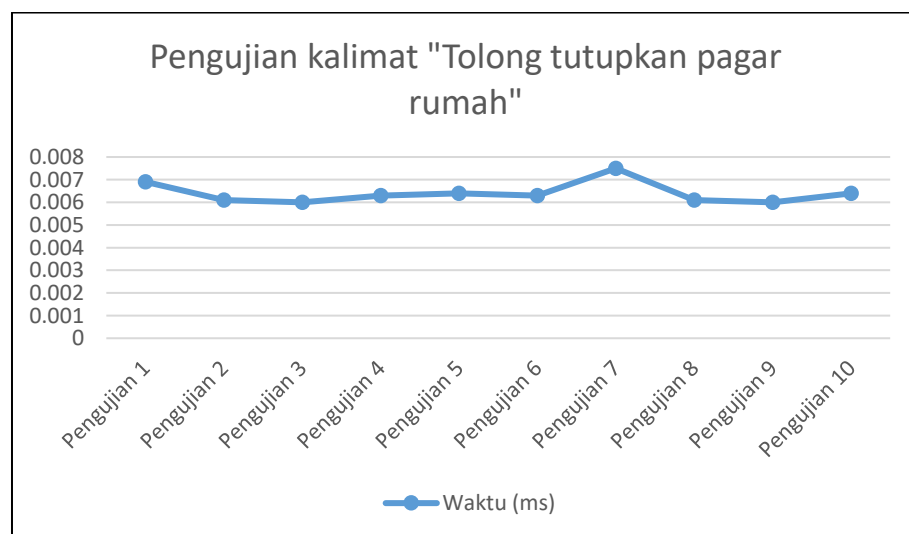
Hasil pengujian kalimat “Tutupkan pagar rumah”.

Kalimat “Tutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0067
2.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0071
3.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0063
4.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0064
5.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0058
6.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0061
7.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0063
8.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0058
9.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0063
10.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0066
Rata-rata					100%	0.00634



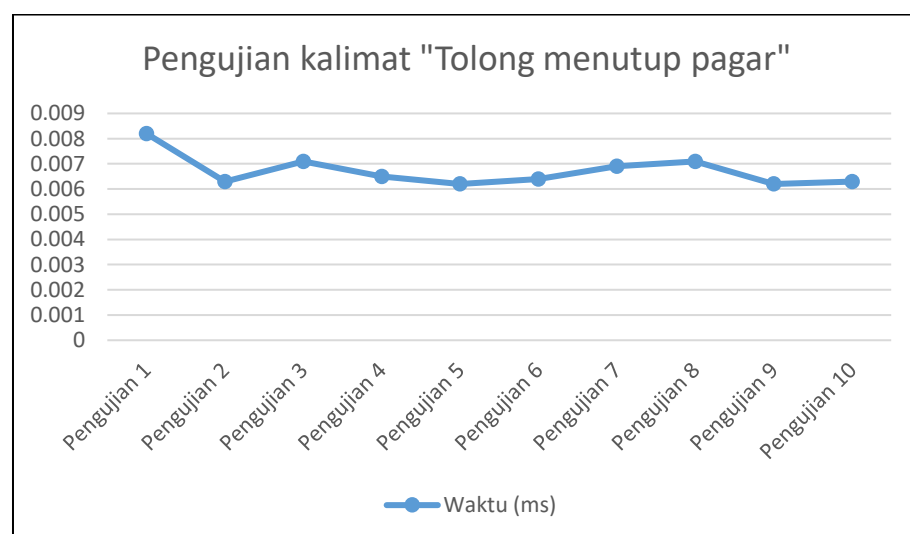
Hasil pengujian kalimat “Tolong tutupkan pagar rumah”.

Kalimat “Tolong tutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0069
2.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0061
3.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.006
4.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0063
5.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0064
6.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0063
7.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0075
8.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0061
9.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.006
10.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0064
Rata-rata					100%	0.0064



Hasil pengujian kalimat “Tolong menutup pagar”.

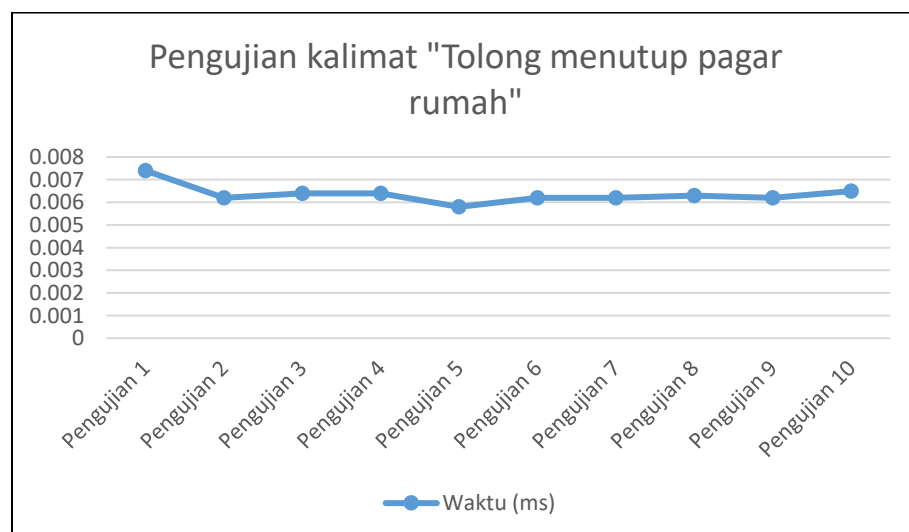
Kalimat “Tolong menutup pagar”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0082
2.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
3.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0071
4.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0065
5.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0062
6.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0064
7.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0069
8.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0071
9.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0062
10.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
Rata-rata					100%	0.00672





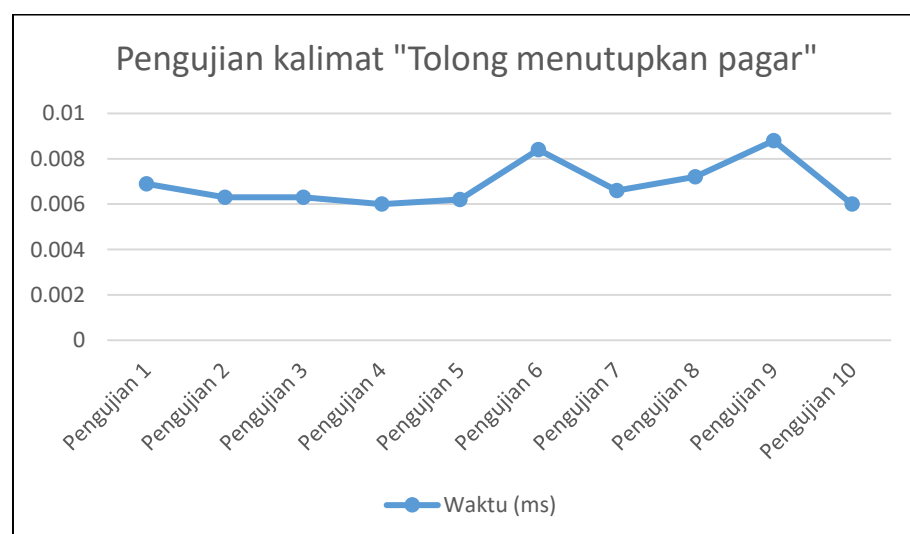
Hasil pengujian kalimat “Tolong menutup pagar rumah”.

Kalimat “Tolong menutup pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0074
2.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
3.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0064
4.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0064
5.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0058
6.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
7.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
8.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0063
9.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
10.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0065
Rata-rata					100%	0.00636



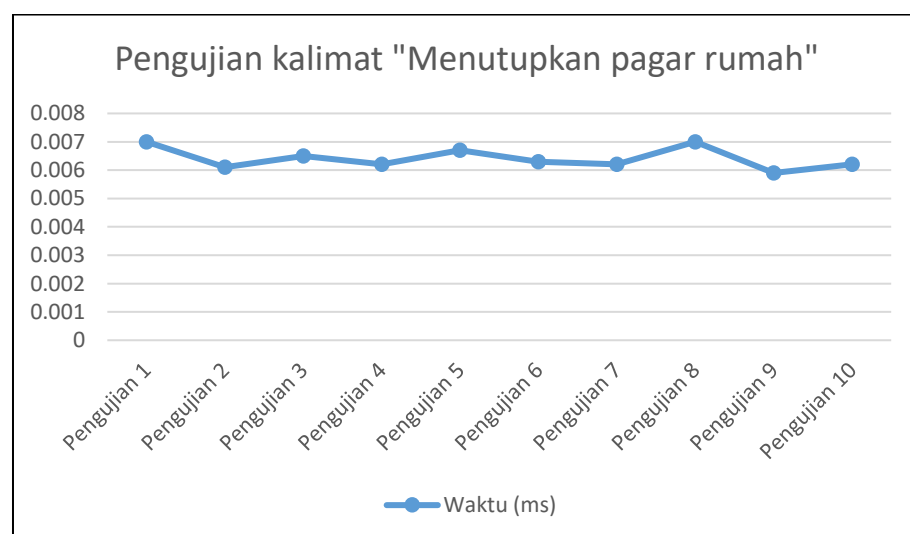
Hasil pengujian kalimat “Tolong menutupkan pagar”.

Kalimat “Tolong menutupkan pagar”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0069
2.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
3.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0063
4.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.006
5.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0062
6.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0084
7.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0066
8.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0072
9.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.0088
10.	Tolong tutup pagar	Tutup pagar	7 Karakter	Tepat	100%	0.006
Rata-rata					100%	0.00687



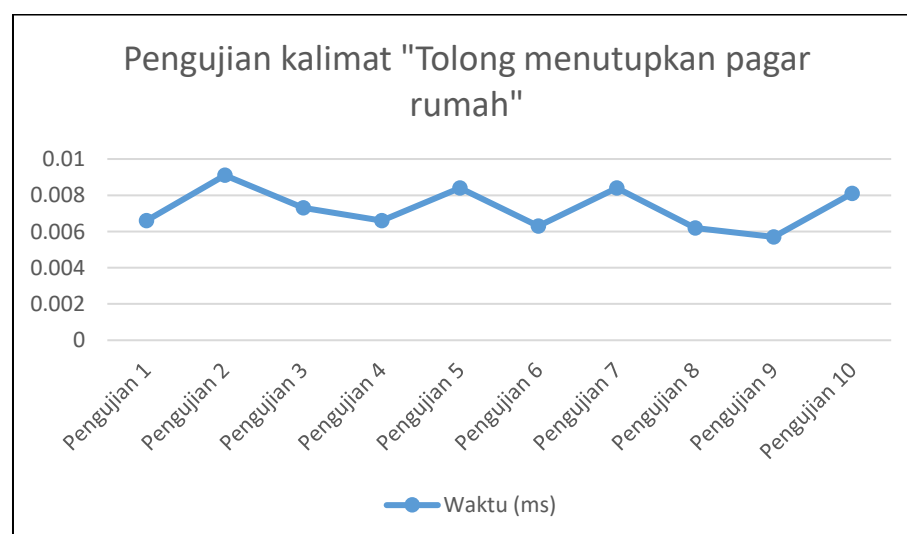
Hasil pengujian kalimat “Menutupkan pagar rumah”.

Kalimat “Menutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.007
2.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0061
3.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0065
4.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
5.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0067
6.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0063
7.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
8.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.007
9.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0059
10.	Tutup pagar rumah	Tutup pagar	0 Karakter	Tepat	100%	0.0062
Rata-rata					100%	0.00641



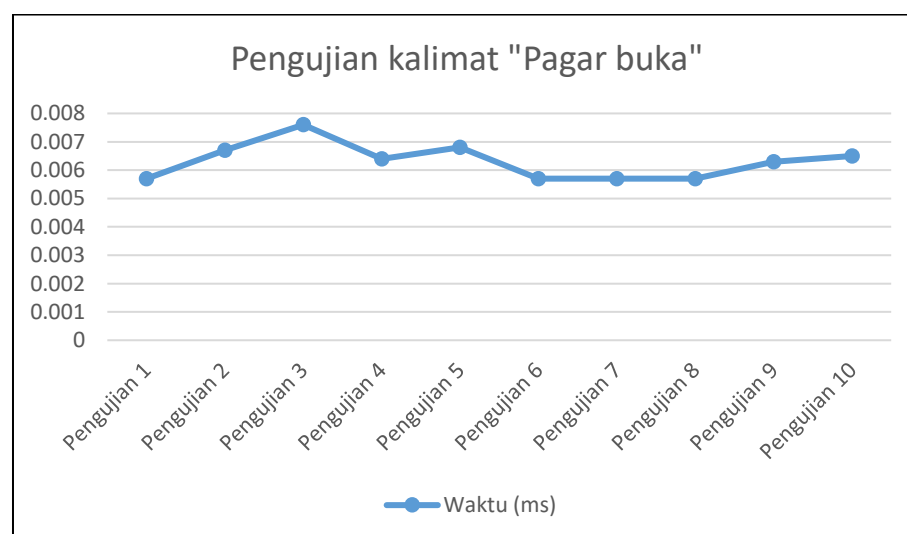
Hasil pengujian kalimat “Tolong menutupkan pagar rumah”.

Kalimat “Tolong menutupkan pagar rumah”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0066
2.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0091
3.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0073
4.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0066
5.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0084
6.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0063
7.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0084
8.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0062
9.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0057
10.	Tolong tutup pagar rumah	Tutup pagar	7 Karakter	Tepat	100%	0.0081
Rata-rata					100%	0.00727



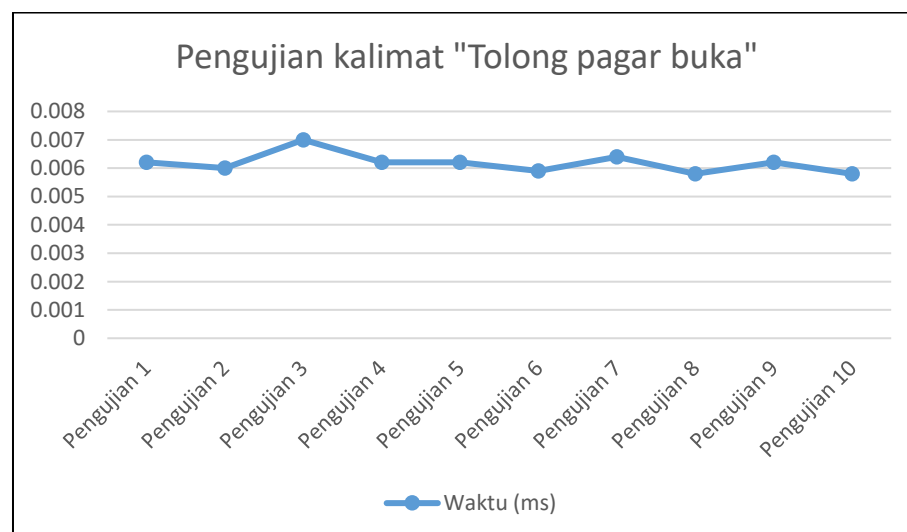
Hasil pengujian kalimat “Pagar buka”.

Kalimat “Pagar buka”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0057
2.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0067
3.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0076
4.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0064
5.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0068
6.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0057
7.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0057
8.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0057
9.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0063
10.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0065
Rata-rata					100%	0.00631



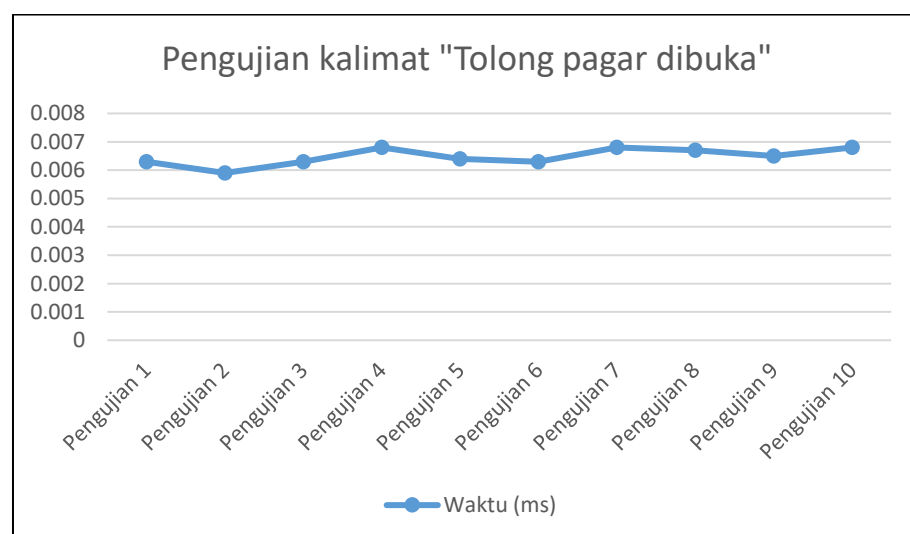
Hasil pengujian kalimat “Tolong pagar buka”.

Kalimat “Tolong pagar buka”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0062
2.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.006
3.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.007
4.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0062
5.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0062
6.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0059
7.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0064
8.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0058
9.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0062
10.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0058
Rata-rata					100%	0.00617



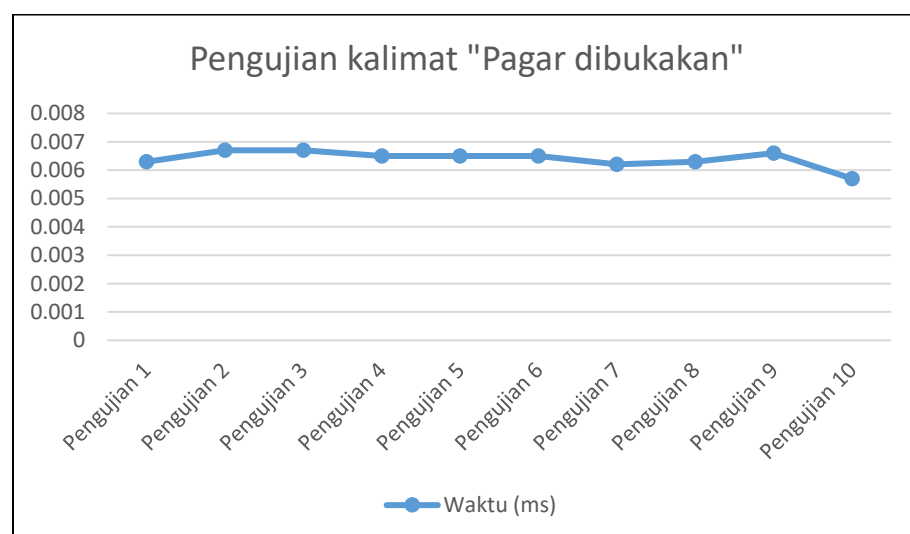
Hasil pengujian kalimat “Tolong pagar dibuka”.

Kalimat “Tolong pagar dibuka”						
No.	Stemming	Pattern	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0063
2.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0059
3.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0063
4.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0068
5.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0064
6.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0063
7.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0068
8.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0067
9.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0065
10.	Tolong pagar buka	Pagar buka	7 Karakter	Tepat	100%	0.0068
Rata-rata					100%	0.00648



Hasil pengujian kalimat “Pagar dibukakan”.

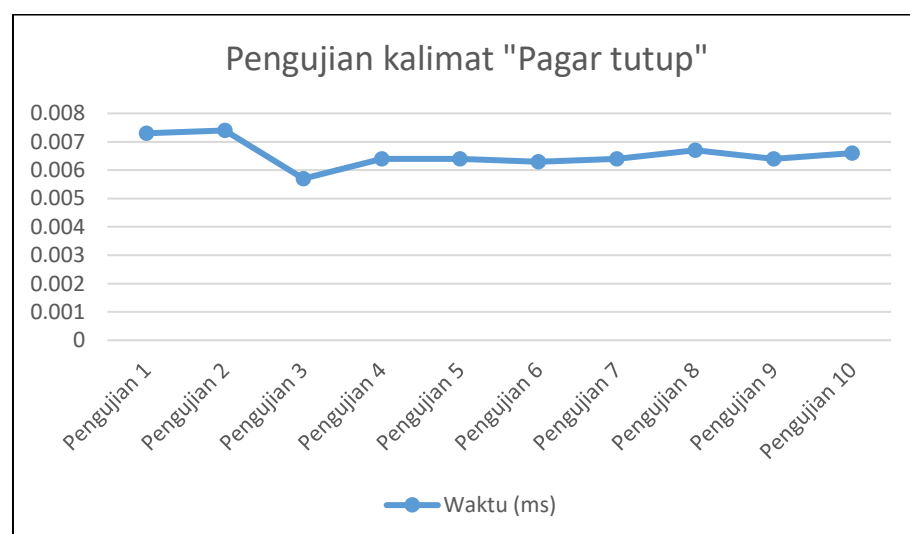
Kalimat “Pagar dibukakan”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0063
2.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0067
3.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0067
4.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0065
5.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0065
6.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0065
7.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0062
8.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0063
9.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0066
10.	Pagar buka	Pagar buka	0 Karakter	Tepat	100%	0.0057
Rata-rata					100%	0.0064





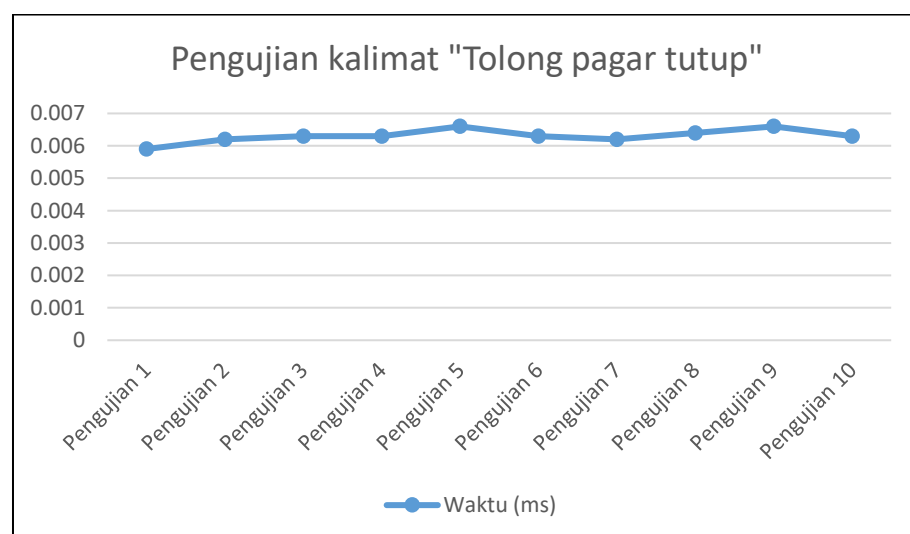
Hasil pengujian kalimat “Pagar tutup”.

Kalimat “Pagar dibuka”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0073
2.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0074
3.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0057
4.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0064
5.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0064
6.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0063
7.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0064
8.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0067
9.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0064
10.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0066
Rata-rata					100%	0.00656



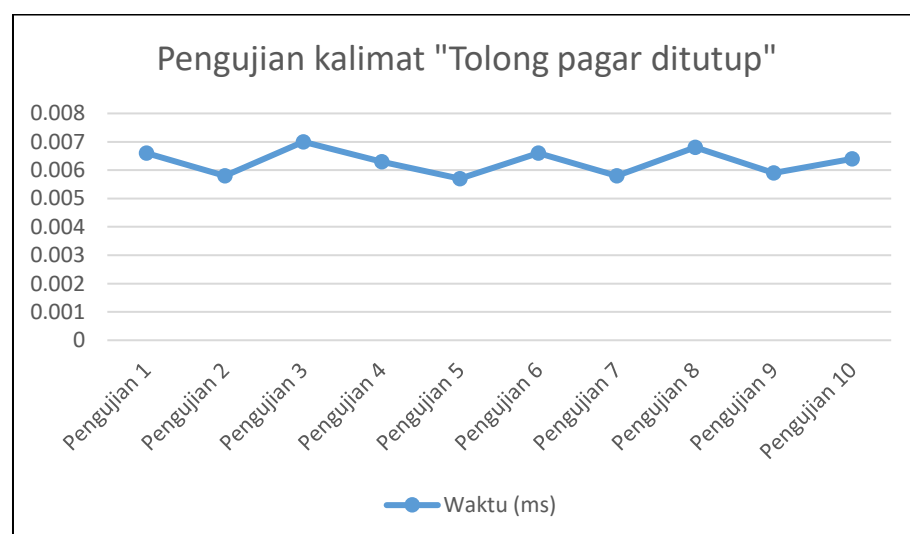
Hasil pengujian kalimat “Tolong pagar tutup”.

Kalimat “Tolong pagar tutup”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0059
2.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0062
3.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0063
4.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0063
5.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0066
6.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0063
7.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0062
8.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0064
9.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0066
10.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0063
Rata-rata					100%	0.00631



Hasil pengujian kalimat “Tolong pagar ditutup”.

Kalimat “Tolong pagar ditutup”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0066
2.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0058
3.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.007
4.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0063
5.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0057
6.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0066
7.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0058
8.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0068
9.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0059
10.	Tolong pagar tutup	Pagar tutup	7 Karakter	Tepat	100%	0.0064
Rata-rata					100%	0.00629



Hasil pengujian kalimat “Pagar ditutupkan”.

Kalimat “Pagar ditutupkan”						
No.	<i>Stemming</i>	<i>Pattern</i>	Pergeseran	Keterangan	Akurasi	Waktu (ms)
1.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0065
2.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0059
3.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.006
4.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0062
5.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0067
6.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0065
7.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0068
8.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0061
9.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0068
10.	Pagar tutup	Pagar tutup	0 Karakter	Tepat	100%	0.0062
Rata-rata					100%	0.00637

